

Amatérské RADIO

NOSITEL
VÝZNAMNÉ
ZA BRANNOU
VÝCHOVU
I. a II. STUPNĚ



mikroelektronika

MIKROPROCESOROVÁ
A VÝPOČETNÍ TECHNIKA
HARDWARE & SOFTWARE

Obsah:

Svazarm v předsjezdovém období	1
Tendence ve vývoji osobních mikropočítačů	2
Postavte si mikropočítač programově kompatibilní se ZX-Spectrum	4
Úprava desky AND-1 SAPI-1	13
Modulová sestava pro mikropočítač ZX-Spectrum	14
Připojení magnetofonu SM260 jako paměťové jednotky k SAPI	24
Osmimístný zobrazovač se sériovým stykem	25

SOFTWARE (programy ze soutěže Mikrogram)

Profesor	30
Testedit	34
Grafický rozhodovací systém	39
Města	43
Slovník	45
Kreslič	47

Programátor paměti PGM-2	53
Kapky matematik v elektrotechnice	60

Text ke II. a III. straně obálky	66
----------------------------------	----

Připojení tiskárny C2111 a C2112 k osobnímu mikropočítači ZX-Spectrum	67
Mikrotermínál MT-80	71
Přerušovací systém a časovač mikropočítačů	74
Tiskárna k SAPI-1 — snáze	76

Amatérské radio příloha „Mikroelektronika“

Vydává ÚV Svazarmu, Opletalova 29, 116 31 Praha 1, tel. 22 25 49 ve Vydavatelství NASE VOJSKO, Václavské náměstí 28, 113 66 Praha 1, tel. 26 06 51-7. Sériový redaktor: Ing. Jan Klábal, OK1UKA, zástupce: Luboš Kalousek, OK1FAC. Redakční rada: předseda Ing. J. T. Hyán, členové: RNDr. V. Brumhofer, OK1HAQ, V. Brzák, OK1DOK, K. Donát, OK1DY, Ing. O. Filippi, V. Gazda, A. Glanc, OK1GW, pplk. Ing. F. Hanáček, P. Horák, Z. Hradský, J. Hudec, OK1RE, Ing. J. Jaroš, Ing. J. Kolmer, Ing. F. Králík, RNDr. L. Kryška, CSc., J. Kroupa, V. Němec, Ing. O. Petráček, OK1NB, Ing. Z. Prošek, Ing. F. Smolík, OK1ASF, Ing. E. Smutný, pplk. Ing. F. Šimák, OK1FSL, Ing. M. Šrámek, OK1NL, doc. Ing. J. Vachek, CSc., laureát st. ceny KG, J. Vortlíček. Redakce: Jungmannova 24, 113 66 Praha 1, tel. 26 06 51-7, Ing. J. Klábal, OK1UKA, I. 354, L. Kalousek, OK1FAC, Ing. P. Engel, A. Hoffmann, I. 353, Ing. A. Myslík, OK1AMY, P. Havlík, OK1PFM, I. 348, sekretariát T. Trnková, I. 355. Rozšiřuje PNS, objednávky do zahraničí vyřizuje rovněž PNS — ústřední expedice a dovoz tisku, závod 01, administrace vývozu tisku, Kašpova 9, 160 00 Praha 6. V jednotkách ozbrojených sil rozšiřuje Vydavatelství NASE VOJSKO, administrace, Václavské náměstí 28, 113 66 Praha 1, tel. 26 06 51-7, I. 294.

Za původnost a správnost příspěvku ručí autor.

Cena jednoho výtisku 10 Kčs.

© Vydavatelství NASE VOJSKO, Praha.

svazarm v předsjezdovém období

Ing. Jan Klábal

V průběhu letošního roku vyvrcholí pětileté úsilí orgánů a organizací za splnění závěrů VII. a příprava na VIII. celostátní sjezd Svazarmu. Celá kampaň výročních besed, konferencí a sjezdů bude prostředkem kritického hodnocení výsledků práce Svazarmu, jeho podílu na plnění závěrů XVII. sjezdu KSČ rozpracovaných 6. společným zasedáním ÚV a RUV Svazarmu a příležitostí k vytyčení dalších záměrů v rozvoji branné výchovné činnosti.

Plnění nových úkolů bude ovlivněno významnými celospolečenskými procesy vyplývajícími z realizace strategie urychlení sociálně ekonomického rozvoje společnosti na základě přestavby hospodářského mechanismu, dalšího rozvoje socialistické demokracie a formování nového myšlení, odpovídajícího intenzivnímu typu rozvoje společnosti i Svazarmu.



Hlavní úsilí zaměřuje celá organizace v současném období na realizaci a objektivní hodnocení účinnosti plnění úkolů politickovýchovné práce vyplývajících z rezoluce VII. sjezdu a zejména ze 4. a 6. zasedání ÚV Svazarmu jako součásti přípravy a průběhu výročních jednání a předsjezdové kampaně. Zvláštní pozornost musí být zároveň věnována stavu a určujícím tendencím branného vědomí, názorům a postojům členů i ostatních účastníků branných činností k vojenské a branné politice ve smyslu závěrů XVII. sjezdu KSČ a požadavku urychlení branné výchovného působení.

V plnění usnesení 8. společného zasedání ústředních výborů Svazarmu a důslednou realizaci Plánu spolupráce ČSLA a Svazarmu se cílevědomě zvyšuje výslednost přípravy branců, vybraných záloh, branců náhradní služby a příprava k civilní obraně.

Ve smyslu závěrů celostátního aktivu klubů důstojníků a praporčíků v záloze je současné období poznamenáno vyšším stupněm organizovanosti DPZ, rozvojem činnosti a zapojením v branné výchově mládeže, přípravě branců a činnosti v civilní obraně. Aktivněji je rozvíjena zájmová činnost KDPZ.

Při rozvíjení zájmové branné činnosti v duchu 7. zasedání ÚV Svazarmu k završení plnění rezoluce VII. sjezdu k masovějšímu a účinnějšímu branné výchovnému působení organizace je řídícími orgány Svazarmu doporučeno

důsledně vycházet z obsahu schválených koncepcí odborností. Za součást obsahu každé odbornosti musí být považována dlouhodobá příprava mládeže pro službu v ČSLA a to jak politická, tak psychofyzická a technická. Musí být dále rozvíjena základní branná příprava, výchova na tradicích. V celém obsahu zájmové branné činnosti ve všech odbornostech při zvyšování účinnosti branně sportovní a branně technické činnosti, výcvikové, tréninkové, informační a propagační činnosti je třeba využívat moderní techniky, zejména elektroniky.

K tomu musí iniciativně krajské, okresní a obvodní výbory organizovat součinnost s dalšími společenskými organizacemi, využívat činnost komisí pro brannou výchovu a komisí pro mládež a tělovýchovu. Musí také rozvíjet metodickou funkci vybraných ZO Svazarmu a jejich klubů vůči školám a dalším společenským organizacím, organizovat a podporovat soutěže zaměřené na rozvoj zájmové branné činnosti mezi neorganizovanou mládeží.

V organizaci samé je pak třeba, jak zdůraznilo 7. zasedání, daleko rozhodněji překonávat různorodou spolkovou omezenost a vytváření bariér. Místo uzavřeného klubismu, který se ještě u některých odborností v nemalé míře projevuje, by měla urychleně nastoupit dělná spolupráce s jinými odbornostmi. Příkladně s těmi, které mají obdobný charakter zájmové činnosti, případně u kterých se mohou vzájemně využívat materiálové prostředky.

Zájmovou brannou činností prolíná polytechnická výchova, ve které je nanejvýš žádoucí v ještě větší míře využívat progresivní techniky a odborně zajištěné technické propagandy. Ve všech činnostech by dále měla být ještě důsledněji spojována odborná činnost s výchovou k tvůrčí práci. Bez progresivní techniky a tvůrčích přístupů nelze do budoucna rozvíjet na vyhovující úrovni většinu odborných činností. Proto se také musí výrazněji v všech odbornostech projevit rostoucí význam uplatňování vědeckých, zejména technických poznatků i zkušeností. V jejich šíření, v technicky odborné propagandě, v politickovýchovné práci a v zájmové činnosti vůbec, mnoho užitečného průběžně zabezpečují svazarmovské odborné časopisy.

Postupné prosazování technického rozvoje v oblasti modernizace provozně materiálové technického zabezpečení a zejména jeho vlivu na plánování, stejně jako úsilí o prosazení zkvalitnění péče o materiál, jeho údržbu a provozuschopnost i odpovídající uložení materiálu, jeho ochranu, předcházení škodám či neoprávněnému využívání a obohacování je dnes jedním z prvořadých úkolů v celé organizaci.

Rovněž v procesu plánování, rozpočtování, vyžadování a využívání finančních, materiálních a investičních

prostředků je právě v předsjezdovém období třeba prosazovat opatření k vysoké hospodárnosti a realizaci schváleného rozpočtu.

V řídicí a organizační práci, ve výstavbě organizace je třeba důsledně prosazovat a realizovat závěry 6. společného zasedání, hlouběji analyzovat situaci a vytvořit předpoklady pro plnění usnesení a úkolů. Funkcionáři a pracovníci na všech stupních musí důsledněji uplatňovat živý styk s nižšími orgány.

Novým prvkem v rozvoji organizační výstavby je požadavek jejího intenzivního rozvoje, čili nepovažovat počet a nárůst základních organizací za rozhodující kritérium a zamezit administrativnímu přístupu k organizační výstavbě a formálnímu náboru členů. Pokud se mají zakládat nové organizace tak jen tam, kde jsou pro to skutečně кадровé podmínky a tam, kde je potřeba rozšiřovat branné výchovné působení.

V samotné členské základně jsou pak současné požadavky ve Svazarmu, vyslovené i na 9. zasedání, směřovány na dosahování vyšší kvality práce. V základních organizacích by již neměly být prvotním kritériem jen počty členů, ale především přístupy k vlastní činnosti a jejímu zkvalitnění se zaměřením na celospolečenský význam a prospěch i výchovné působení a to nejen na jednotlivé členy, ale i do širšího okolí. Určitou výjimkou jsou zde elektronické odbornosti, převážně z oblasti mikroelektronických aplikací (výpočetní techniky, mikroprocesorů, automatizovaných a robotizovaných obvodů a systémů), kde je žádoucí tyto dále rozvíjet na vyšší úrovni a podle daných možností je i materiálně technicky co nejlépe zabezpečit (případně i z dovozu), a docílit tak ještě vyššího zájmu o tuto branně technickou činnost zejména mezi mládeží. Je to nanejvýš nutné nejen z hlediska zabezpečení dalších odborných kadrů pro národní hospodářství, ale hlavně pro potřeby brannosti, kde je třeba uvažovat o obsluze nejmodernější vojenské techniky vybavené elektronikou, automatizovanými a mikroprocesorovými soustavami. To také vyplývá ze základních úkolů Svazarmu, i nadále v nadcházejícím období provádět důslednou přípravu branné výchovné aktivity pro zabezpečení úkolů ve prospěch čs. ozbrojených sil a zabezpečit další rozvoj zájmové branné činnosti, zejména na školách.

Branné a branně technické sporty rozvíjí Svaz pro spolupráci s armádou jako prostředek masového branné výchovného působení. Sleduje jejich důsledné zaměřování k potřebám socialistického způsobu života, k formování morálně politických, odborně technických a fyzických vlastností a schopností uvědomělého budovatele a obránce socialistické vlasti.

Svaz pro spolupráci s armádou jako společenská dobrovolná organizace má při budování a obraně socialistické společnosti nezastupitelné postavení. Plněním úkolů jednotného systému branné výchovy obyvatelstva a svými zájmovými branně technickými a branně sportovními činnostmi přispívá k rozvoji branné společenského života občanů našeho socialistického státu.

TENDENCE VE VÝVOJI OSOBNÍCH MIKROPOČÍTAČŮ

Ing. Branislav LACKO, CSc.
Výpočetní středisko TOS Kuřim

Jak vyplývá ze statistik i z prognostických analýz je oblast osobních mikropočítačů velmi dynamicky se rozvíjejícím odvětvím jak v zahraničí tak i u nás.

Tento rozvoj zasáhne ve svých důsledcích všechny oblasti společenského i hospodářského života. Rozšíření osobních mikropočítačů je přímým důsledkem probíhajícího procesu elektronizace v našem národním hospodářství. Kromě jiného vyžaduje, aby se i v průběhu vzdělávacího procesu na školách každý žák seznámil s principy práce počítače a se základy algoritmizace, programování a obsluhy. Často se mluví o „druhé gramotnosti“, což jsou nezbytné vědomosti budoucího (už i současného) člověka — být schopen používat počítače jako nástroj ve své každodenní praxi.

Poznamenejme, že tento prudký vývoj má dopad i do dalších oblastí. Např. jedno americké vydavatelství vydalo jen v roce 1983 300 knih o osobních počítačích a v roce 1984 dalších 400 titulů. V NSR vycházelo v roce 1985 na 45 časopisů, zabývajících se hlavně programovým vybavením a aplikací mikropočítačů.

K lepší orientaci výchovy žáků a studentů na školách přispěje jistě i zamyšlení nad dalším vývojem osobních mikropočítačů, protože naše budoucí generace by měla být připravena na takové prostředky výpočetní techniky, které bude používat v pozdější praxi, nikoliv na ty, které se v té době již nebudou vyrábět.

Základní pojmy a rozdělení mikropočítačů

Rychlý rozvoj mikropočítačů přivádí s sebou i řadu negativních faktorů. Jedním z nich jsou terminologické potíže v důsledku zaostávání tvorby pojmů za rychlým technickým vývojem.

Proto jsou zde uvedeny základní pojmy a přehledné rozdělení mikropočítačů s cílem vytvořit jednotnou pojmovou základnu alespoň pro účely tohoto článku.

Mikropočítačem rozumíme číslicový samočinný počítač realizovaný na bázi mikroprocesoru, u jehož návrhu a realizace byla uplatněna jako rozhodující hlediska minimální ceny a minimální velikosti, při zachování dostatečných funkčních vlastností pro potřeby uživatelů.

U střediskových počítačů všech velikostí (malých, středních, velkých) a u superpočítačů jsou naopak rozhodujícími hledisky velikosti výkonu, kapacity paměti a počet výkonných periférií. Mikropočítač tvoří funkčně samostatný celek na rozdíl od zabudovaných mikroprocesorů, které jsou nedílnou součástí strojů a motorů. Mikropočítač je uživatelsky aplikovatelný, což znamená, že obsahuje všechna zařízení potřebná pro aplikaci a nutné programové vybavení.

Mikroprocesor je integrovaný obvod velké nebo velmi velké integrace, který realizuje podstatnou část funkcí základní jednotky počítače.

Programový kalkulátor je zařízení založené na mikroprocesoru a orientované na zpracování numerických výpočtů zadáváním převážně lineárních algoritmů. Programování je prováděno zapa-

matování posloupností zmáčknutím tlačítek popř. znakovým symbolickým zápisem zadávaných výrazů. Od mikropočítače se liší tím, že nepoužívá vyššího programovacího jazyka a jeho programy nejsou samomodifikující.

Mikropočítače lze dělit na skupiny podle různých hledisek.

Podle typu uživatele

- **Profesionální mikropočítače**, které jsou používány školenými pracovníky při výkonu svého povolání.
- **Mikropočítače pro volný čas**, které jsou používány pro zábavu a rekreační programování.

Podle účelu použití dělíme profesionální mikropočítače na:

- **mikropočítače pro vědecko-technické výpočty**, orientované na řešení numerických problémů, které jsou často doplňovány prostředky pro grafický vstup a výstup;
- **kancelářské mikropočítače** pro automatizované zpracování dat v administrativě, jež zahrnuje:
 - vstup a uložení dat;
 - decentralizované zpracování dat;
 - zpracování textů
 (na tyto operace mohou být některé kancelářské mikropočítače specializovány);
- **řídicí mikropočítače** pro automatizované řízení rozličných technologických procesů;
- **mikropočítačové vývojové systémy**, určené pro vývoj programového vybavení mikropočítačů a mikroprocesorů;
- **školní mikropočítače** využívané u předmětů z výpočetní techniky;
- **speciální mikropočítače** (např. mikropočítače pro kosmický výzkum, vyučovací stroje).

Mikropočítače pro volný čas lze podle účelu použití rozdělit na dvě velké skupiny:

- specializované mikropočítače na televizní hry;

- domácí počítače pro víceúčelové použití.

Podle **způsobu provedení** dělíme mikropočítače na:

- stolní;
- přenosné (speciálně upravené pro převážení);
- kapesní.

Na základě **způsobu výroby** rozlišujeme:

- sériově vyráběné mikropočítače;
- mikropočítače sestavené na objednávku z průmyslově vyráběné stavebnice mikropočítačového systému;
- amatérsky navržené a sestavené mikropočítače;
- mikropočítače sestavené amatérsky z průmyslově vyráběné stavebnice mikropočítače.

Podle **šířky toku dat** na sběrnici rozdělujeme mikropočítače s šířkou toku dat.

- 8 bitů;
- 16 bitů;
- 32 bitů.

Podle **architektury** dělíme mikropočítače na:

- mikropočítače se společnou sběrnici;
- s vlastní архитектурou; (data flow, maticové procesory apod.).

Osobní mikropočítače vyčleňujeme z profesionálních mikropočítačů a z mikropočítačů využívaných ve volném čase jako mikropočítače k individuální potřebě.

Názvem se chce zdůraznit:

- že tento mikropočítač obsluhuje a využívá jedna a tatáž osoba a není potřeba, aby provozování zajišťoval tým specialistů;
- že uživatel má zdroje celého mikropočítačového systému při výpočtu zcela k dispozici a nesdílí je s jinými uživateli.

Poznámky

1. U mikropočítače je rozhodující nízká cena a malé rozměry. Výkonnost je zde relativní pojem podmíněný historicky. Tak, jak se s rozvojem výpočetní techniky zvyšují její technické parametry, může být určitý výpočetní výkon považován za velký, popř. malý v rozdílném časovém období.
2. Rozvoj výpočetní techniky rozšiřuje rozdělení i počty kategorií u jednotlivých hledisek. Proto i navržené rozdělení je nutno chápat jako pracovní.
3. Jednou z obecných vlastností mikropočítačů je univerzálnost. Proto jednotlivé kategorie mikropočítačů je nutno chápat jako neostré. (Na mikropočítači pro VTV lze rovněž zpracovávat texty a kancelářský mikropočítač pro decentralizované zpracování dat může výsledky prezentovat graficky).

Vývoj mikropočítačů a pamětí

Mikroprocesor je základním stavebním kamenem každého mikropočítače a vtiskuje příslušnému mikropočítači

Základní charakteristiky mikropočítačů

Rok	1979	1985	1990 (výhled)
architektura (mikroprocesory)	1x8 bit	1x16b + speciální koprocessory	2—10x16/32b
režim práce	monoprogramový	multiprogramování	multiprocessing
objem operační paměti RAM	32 až 64 kB	126 až 640 kB	512 kB až 4 MB
ROM	2 až 4 kB	32 kB	128 kB
Operační systém	CP/M	MS-DOS	UNIX
Programovací jazyky	60% BASIC	40% Pascal	35% Ada
	20% Assembler	25% Basic	35% C
	20% ostatní	20% ostatní	10% Pascal
		5% assembler	10% ostatní
grafika	—	640x200 barevně	1024x1024 barevně
znaky	(256x92) 20ř x 32 zn	25ř x 80 zn barevně	25ř x 132 zn barevně
tisk	mozaiková tiskárna	mozaiková tiskárna s grafickými možnostmi	barevná laserová tiskárna
vnější paměť	2x8" disketa à 256 kB	5,25" disketa 360 kB + 5,25" pevný disk 20 MB	3,5" disketa 2,5 MB + 5,25" pevný disk 140 MB + LASER RAM s kapacitou 0,5 GB
telekomunikační možnosti	—	lokální síť LAN	heterogenní síť s dálkovým přenosem dat
operační rychlost	25 000 op/sec	0,5 MIPS	10 MIPS

charakteristické vlastnosti a určuje jeho vnitřní architekturu.

Dynamický rozvoj výroby mikroprocesorů dokumentuje následující skutečnost: první mikroprocesor Intel 4004 se objevil v roce 1971, ale již v roce 1981 použil americký průmysl 500 000 mikroprocesorů a v celém světě bylo 1 milión přístrojů se zabudovaným mikroprocesorem.

Ruku v ruce s vývojem mikroprocesorů postupuje i výroba polovodičových pamětí. Firma IBM (USA) začala od letošního roku montovat do svých výrobků tolik očekávané megačipy — paměti s kapacitou 1 milión bitů na jedné křemíkové destičce 5x5 mm. Výrobu pamětí o kapacitě 4 Mb plánuje firma Siemens v novém závodě v Pullachu již v roce 1989.

Zatím ovládají oblast polovodičových pamětí japonští výrobci s produkci laciných integrovaných pamětí 256 kb.

V roce 1988 se očekává osobní mikropočítač označovaný „3M“ (1 MB — 1 MIPS — 1 Mpixel) a supermikropočítač označovaný „10-10-10“ (10 MB — 10 MIPS — 10 000 ř).

Hlavní směry rozvoje osobních mikropočítačů

Přechod na mikroprocesory 32b a 64b.
Zvětšení kapacity operační paměti nad 1 MB.
Využití přehrávačů CD, DCI, DAT, VIDEO DISC, paměti OROM, ORAM.
Zvětšené komunikační možnosti.
Vstup a výstup mluveným slovem.
Práce na více úlohách současně.
Nástup programovacího jazyka Ada.

Rozšíření operačního systému UNIX. Zapojení osobních počítačů do integrovaného zpracování dat a automatizované kanceláře.

Využívání výsledků výzkumů umělé inteligence (expertní systémy, systémy pro podporu rozhodování — DDS, CAD + CAM + CAP + CAT = CIM, CAI apod.)

Programovací jazyky

Jazyk symbolických adres pro všechny typy mikroprocesorů.

Dosavadní známé jazyky podle ANSI:

- a) generování kódu; COBOL 74, FORTRAN 77, Pascal, PL/M, Ada;
- b) interpretačním systémem; BASIC (někdy také jako kompilátor), APL

Nové jazyky

C — systémový jazyk vyvinutý v Laboratořích firmy BELL rozšířený díky skutečnosti že operační systém UNIX je napsán v tomto jazyku. Jedná se o dost univerzální programovací jazyk.

FORTH — vznikl v roce 1970 pro potřeby řízení teleskopu v USA. Po jeho úspěšném nasazení vznikla samostatná firma FORTH, která jazyk zdokonalila a nabídla k implementaci na nejrozšířenější mikroprocesory 8b (Z80, i 8080, i 8085) i 16b (i 8086,

M68000, Z800). Jedná se interpretační jazyk orientovaný na práci se zásobníkem. Je použito notace RPN pro zápis výrazů.

LOGO — původně určený pro výuku programování žáků základních škol v USA, byl přepracován na MIT pro jazyk orientovaný na snadné vytváření grafických obrázků ať už z klávesnice nebo pomocí ukazovacího zařízení.

Smalltalk — původně vyvinut pro mikropočítače Xerox. Využívá principů modulárního programování a souboru typizovaných podprogramů. Jsou v něm uplatněny principy neprocedurálního programování (co se má provádět, ne jak se to má provádět).

PROLOG — jazyk vybraný japonským projektem 5. generace počítačů jako základní jazyk pro řešení úloh umělé inteligence. Je založen na t.z.v. Hornovských klauzulích (podřídka logiky prvního řádu a Robinsonové rezolučním principu).

LISP — jazyk na zpracování seznamů s využitím LAMBDA kalkul; jazyk původně určený na zpracování nenumernických úloh se s výhodou používá pro řešení úloh umělé inteligence.

Operační systémy

Z různých operačních systémů se prosadily ve větší míře následující:

CP/M (Digital Research)

Prakticky mezinárodní standard 8b mikropočítačů na mikroprocesorech i 8080, i 8085, Z80 (odhaduje se asi 800 tis. instalací) velmi univerzální a portabilní. Srovnatelný ekvivalent v ČSSR MIKROS, v NDR SCP. Obsluha jednoho uživatele bez multiprogramování. Podmínky pro aplikaci: alespoň 56 kB RAM, dvě disky po 126 kB, klávesnice ASCII, obrazovkový alfanumerický displej s kurzorem a mikrotiskárna.

ISIS (Intel)

Operační systém rozšířený díky mikropočítačovým vývojovým systémům firmy Intel pro mikroprocesory i8080 a i8085. Práce s jedním programem. Lze ho použít i pro i8088.

REX 80 (Systems and Software)

Operační systém pro 8b mikroprocesory i8080 a Z80 pracující v aplikacích reálného času. Dodává se včetně zdrojového textu.

PC-DOS (IBM, původně MS-DOS firmy Microsoft)

Téměř mezinárodní standard v oblasti 16b personálních počítačů s mikroprocesorem i8086 v důsledku jeho použití u IBM PC.

CP/M-86 (Digital Research)

Již ne tak úspěšná replika CP/M od téže firmy pro multiprogramovací prostředí až 16-ti úloh pro 16b mikroprocesory i8086 (pro mikroprocesory M 68000 je ve verzi CP/M-86k). Existuje i ve verzi CONCURRENT CP/M-86 pro aplikaci v reálném čase.

UNIX-Microsoft (prodej) — Bell Telephone Lab. (vývoj) Multiprogramovací operační systém 16b mikropočítačů a mikropočítačů PDP11, DEC-VAX a dalších. Umožňuje i použití 32b mikroprocesorů. Novější verze je označovaná Xenix.

POSTAVTE SI MIKROPOČÍTAČ PROGRAMOVĚ KOMPATIBILNÍ SE ZX-SPECTRUM

Ing. Aleš Juřík

Na našem trhu se v poslední době objevilo několik typů mikropočítačů, o které je stále velký zájem. Jsou také cenově přístupné širší veřejnosti. Svědčí o tom zejména rychlost, s jakou mizí z pultů prodejen Domácích potřeb i Tuzexu. Nejrozšířenějším typem mikropočítače u nás je v současné době ZX Spectrum firmy Sinclair a následující odvozené typy (ZX Spectrum Plus, Delta, atd.). Na tyto typy mikropočítačů je u nás k dispozici velké množství rozsáhlého programového vybavení, překladačů vyšších programovacích jazyků počínaje a různými typy her konče.

To byl jeden z hlavních důvodů, proč jsem při stavbě amatérského mikropočítače zvolil koncepci programově kompatibilní s tímto profesionálně vyráběným mikropočítačem, přestože jsem si vědom, že jeho obvodová struktura není nejvýhodnější. Neumožňuje totiž bezproblémové rozšiřování systému, a to zejména díky použitému způsobu lineárního adresování periférií. Při návrhu jsem se snažil tyto nevýhody potlačit. Přineslo to s sebou některé programové nekompatibility (byla zjištěna jedna hra, která se dá ovládat pouze ovládačem, a jedna, u které nefungují časové smyčky tak jak na originálním mikropočítači vzhledem k jejich umístění z části ve videoRAM a z části v horní polovině paměti). Z mého hlediska je to nepodstatné, ale přesto bude uveden způsob, jak udělat adresování plně kompatibilní s originálním obvodovým zapojením. Je však třeba zdůraznit, že vzhledem k jiné filozofii zobrazování videoRAM budou odstraněny časové disproporce programů, uložených ve videoRAM. Tyto programy zde probíhají o více než třetinu rychleji. U žádného autorovi dostupného systémového programu nebyla zjištěna nefunkčnost. Při stavbě jsem se snažil důsledně vycházet ze součástkové základny, dostupné počátkem roku 1987 v maloobchodní síti. Přesto bylo nutno použít dvou typů obvodů, které v té době nebyly k dostání, a to mikroprocesoru Z80A, popřípadě jeho ekvivalentu z NDR s označením UA880D, a dynamických pamětí RAM 64kb s dobou přístupu pod 150 ns.

2. Filosofie návrhu

V dalším textu jsou signály aktivní v log. 0 označovány buď čarou nad názvem nebo písmenem N na konci názvu. Signál LOADN je tedy ekvivalentní signálu LOAD. Protože užívání značení žádného signálu nekončí písmenem N, je tato konvence jednoznačná. Jako videoRAM je označována druhá čtvrtina vnitřní paměti (adresy 4000H až 7FFFH). Předtím, než budou popsány jednotlivé obvodové bloky navrhovaného obvodového řešení, bude popsán základní obvod ULA, používaný v originálním Spectru, a to nikoliv z hlediska zapojení vývodů, ale z hlediska funkční filozofie. Většina obvodů totiž zastupuje tento základní IO. ULA zabezpečuje tyto funkce:

a) Tvorba videesignálu:

- I) čítače adresy displeje,
- II) multiplexery adresy displeje pro dynamickou paměť RAM (přepínání adresy jas/atribut),
- III) zajištění pozdržení činnosti procesoru v okamžiku přístupu obvodu do videoRAM (pozastavení hodin),
- IV) obvody, generující řádkové a snímkové synchronizační signály,
- V) obvody, tvořící videesignály Y (jas) a U, V (signály barvy v normě PAL).

b) Připojení základních periférií:

- I) připojení klávesnice,
- II) připojení magnetofonu,
- III) připojení reproduktoru.

c) Vytvoření inicializačního signálu pro samotný obvod.

d) Generování hodinového taktu pro procesor.

e) Přerušování běhu mikroprocesoru s kmitočtem 50 Hz, maskovatelným přerušováním pro zabezpečení periodické programované obsluhy klávesnice a vnitřního počítadla času.

f) Tvorba signálů RAS a CAS pro videoRAM.

ad a-I) Při návrhu by měly být obvody čítačů adresy displeje sestaveny tak, aby generovaly bodový kmitočet 7 MHz (což je zároveň základní kmitočet oscilátoru mikropočítače) a adresu 32 osmic bodů na řádek a 192 řádek na snímek. Základní kmitočet oscilátoru obvodu ULA je sice 14 MHz, ale tento kmitočet není v navrhované koncepci nutný. Zároveň musí být vytvářen signál udávající zda adresa je platná a bude se zobrazovat informace z videoRAM nebo zda se bude zobrazovat jasová informace, na kterou je nastaven registr borderu. Signál borderu zabírá 6/14 délky TV řádku a 8/20 TV pulsů. Z přepínacího signálu a z adresových signálů čítače je vhodné odvodit synchronizační impulsy pro TV videesignál.

ad a-II) dále je třeba přepínat adresu bajtu, v němž je uložena jasová informace o příštích osmi zobrazovaných bodech (udávající, že při log. 1 je zobrazován jas, odpovídající jasnosti a při log. 0 jas, odpovídající jasu

papíru) s adresou bajtu, v němž je uložen odpovídající atribut (bajt, který má na bitech 0 až 2 informaci o barvě inkoustu, na bitech 3 až 5 informaci o barvě papíru, bit 6 obsahuje informaci o přijasnění a bit 7 o blikání). Když si projdeme překlad ROM Spectra, zjistíme, že jasové informace jsou uloženy na adresách 4000H až 57FFH, zatímco odpovídající atributy obsahuje paměť v rozsahu 5800H až 5AFFH. Odpovídající úroveň na příslušných adresách se nastavují podle následujícího rozpisu (ADHx — x-tý bit adresy displeje horizontálně, ADVy — y-tý bit adresy displeje vertikálně):

bit AB	obsah při jasu	obsah při atributu
AB0	ADH0	ADH0
AB1	ADH1	ADH1
AB2	ADH2	ADH2
AB3	ADH3	ADH3
AB4	ADH4	ADH4
AB5	ADV3	ADV3
AB6	ADV4	ADV4
AB7	ADV5	ADV5
AB8	ADV0	ADV6
AB9	ADV1	ADV7
AB10	ADV2	log. 0
AB11	ADV3	log. 1
AB12	ADV7	log. 1
AB13	log. 0 (AVS0)	log. 0 (AVS0)
AB14	log. 1 (AVS1)	log. 1 (AVS1)
AB15	log. 0 (AVS2)	log. 0 (AVS2)

Toto adresování je dosti „složitě“ z toho důvodu, že ULA kvůli minimalizaci časových ztrát vyčítá obrazovou informaci ve stránkovém módu (dolních 8 bitů musí být shodných). Protože navrhované řešení nepozastavuje činnost procesoru (viz dále), není spolupráce s RAM ve stránkovém módu nutná. Přináší to obvodové zjednodušení, neboť přepínání adres snadno zajistíme dvojitými multiplexery. Je vidět, že pokud bychom v době adresování paměti obvodem ULA přivedli na vodiče AB13 až AB15 nikoli konstantní logické úrovně, ale výstupy pomocného třibitového registru AVS0 až AVS2, mohli bychom získat osm oblastí paměti, jejichž obsah by se dal zobrazovat.

ad a-III/ Když prostudujeme časovací diagram procesoru Z80A, zjistíme, že nemůže nastat situace, kdy by požádal o přístup do paměti častěji, než jednou za dvě periody hodinového signálu. Při kmitočtu hodinového signálu 3,5 MHz je perioda 286 ns. Procesor tudíž může pracovat s pamětí maximálně jednou za 572 ns. Je vidět, že pokud bude bodový kmitočet displeje vhodně synchronizován s hodinovým signálem procesoru a k dispozici budou dostatečně rychlé paměti RAM, je možno navrhnout časový multiplex přístupů do paměti tak, aby procesor nemusel být pozastavován. Synchronizace se nejjednodušeji zajistí tak, že od jednoho oscilátoru se bude odvozovat jak bodový kmitočet, tak hodiny procesoru. Tudíž programy umístěné v oblasti videoRAM budou probíhat o něco rychleji, než na originálním mikropočítači, zatímco programy umístěné ve zbytku paměti a nekomunikující s oblastí videoRAM poběží stejně rychle. Časový multiplex musí být však navržen tak, že nadřazenou (prioritní) složkou bude procesor. Informace pro displej musí být vyčítána z paměti

okamžitě, jakmile o přístup do paměti procesor nežádá, a to právě dvakrát za čtyři hodinové cykly. Zobrazení osmi bodů na obrazovce trvá totiž právě čtyři hodinové cykly a pro zobrazení osmi bodů je třeba vyčtení dvou bajtů z videoRAM (jas a atribut). Protože však nevíme, za kolik cyklů bude toto vyčítání skončeno (může být za dva nebo za čtyři), je nutno do cesty obou signálů zařadit po jednom osmibitovém vyrovnávacím registru pro synchronizaci plnění registrů displeje.

ad a-IV/ Základ návrhu těchto obvodů byl popsán v části a-II/, pouze je třeba zajistit sečení řádkových a snímkových synchronizačních impulsů pro získání úplné synchronizační směsi.

ad a-V/ Tato část úzce navazuje na obvody časového multiplexu paměti. Informace o jasu a atributu je jednou za osm period bodového signálu přepsána do registrů displeje. První z nich musí být posuvný registr, který je plněn jasovou informací. Druhý je vyrovnávací paralelní registr. Výstup posuvného registru musí ovládat přepínání signálů jasové informace mezi inkoustem a papírem v závislosti na nastavení příznakového bitu blikání. Dále musí být ovládáno přepínání jasu na jas borderu (např. pomocí signálu SCR). To se dá zabezpečit třemi trivstupovými multiplexery, na jejichž výstupu dostaneme signály RGB. Společně se signálem synchronizační směsí SYNC a signálem přijasnění z nich dostaneme v součtovém analogovém obvodu kompletní videosignál. Tento videosignál můžeme vést do videovstupu televizoru nebo monitoru. Pomocí dalších obvodů můžeme vytvořit signály U, V a posléze i kompletní videosignál v normě PAL (norma SECAM se pro výstup mikropočítače příliš nehodí vzhledem k nejednoznačnosti barevného zobrazení vodorovné čáry o šíři jednoho bodu). Je však třeba uvážit, není-li lepším řešením úprava vstupu barevného televizoru pro signály R, G, B, I (intenzita) a SYNC. Obvodově to vyjde jednodušeji a kvalita zobrazení je lepší. Takto je možno použít i televizory bez dekodéru PAL.

ad b/ Připojení tří základních periférií je velice jednoduché. Všechny tři jsou totiž adresovány jedinou adresou (FEH-254D). Vstupem klávesnice je horních osm bitů adresové sběrnice, výstupem pět bitů, připojovaných vstupní instrukcí IN (C). A na dolních pět bitů datové sběrnice. Tato instrukce posílá na dolních osm bitů datové sběrnice obsah registru C, pomocí kterého je adresována periférie, na horních osm bitů je poslán obsah registru B. Je využíván k postupnému aktivování linek klávesnice. V registru B je totiž log.0 pouze na jednom bitu. Rotací obsahu tohoto registru a periodickým čtením klávesnice se dá zjistit, které tlačítko je zmáčknuto. K oddělení horních osmi bitů adresové sběrnice od klávesového pole stačí použít osm diod, zapojených katodami na sběrnici.

Magnetofon je připojen pomocí dvou bitů — vstupního a výstupního. Signál při nahrávání je čten z bitu D6 datové sběrnice a při ukládání na magnetofon je informace posílána na bit D3. Na bit D4 je posílána informace, ovládající reproduktor a na bitech D0 až D2 aktuální informace v barvě borderu. Potřebuje

me tedy pětibitový registr pro zápis dolních pěti bitů datové sběrnice. Obousměrné oddělení je vhodné (oddělovače sběrnice).

ad c/ Pokud nepoužijeme pevnou paměť ROM 16 kB, je nutno po zapnutí napájení kromě inicializace procesoru ještě zajistit připnutí stínové ROM. V té musí být program, který nahraje do prvních 16 kB paměti RAM obsah originální paměti ROM ZX Spectrum. Výhodou tohoto řešení je to, že nemusíme používat pouze program z originální ROM, ale i program upravený (například opravené chyby originálního programu, obsah paměti ROM, který je pod názvem ISO ROM nabízen v Západní Evropě, popřípadě do volného místa napsat své vlastní programy). Nesmíme však zapomenout na to, že do ROM se nesmí dát zapsat informace (samotná originální ROM zapisuje sama do sebe, mnoho firemních programů zapisuje „přes“ ROM kvůli znesnadnění kopírování takto umělé prodlouženého bloku). Po ukončení nahrávání se musí stínová ROM automaticky odepnout a program znovu spustit od adresy 0000H.

ad d/ Hodinový kmitočet lze nejjednodušeji získat vydělením bodového kmitočtu displeje dvěma; získáme tak přesně 3,5 MHz.

ad e/ Pro generování přerušení lze s výhodou použít snímkové synchronizační impulsy. Při plánovaném dalším použití vstupu INTN je nutno tento signál přivádět přes výstup s otevřeným kolektorem, jinak jej lze připojit přímo. Přerušeni bude sice generováno v jiné fázi vůči zobrazování obrazovky než v originálním Spectru (tam je generováno v době prvního aktivního řádku), to ale nemá praktický význam.

ad f/ Vzhledem k tomu, že je použita fyzicky pouze jedna paměť, je signál RAS generován pro celou paměť jednou. Je třeba odlišit signál RAS při vstupu procesoru do paměti (značeno RASP) a signál RAS, pokud jsou vyčítána data pro displej (značeno RASD). Signál CAS je totiž nutno generovat při každém přístupu displeje do paměti, zatímco při obnovovacím cyklu procesoru se generovat nemusí. Ovládáním signálu CAS lze také jednoduše připojit expanzní paměť RAM.

Obvodové řešení

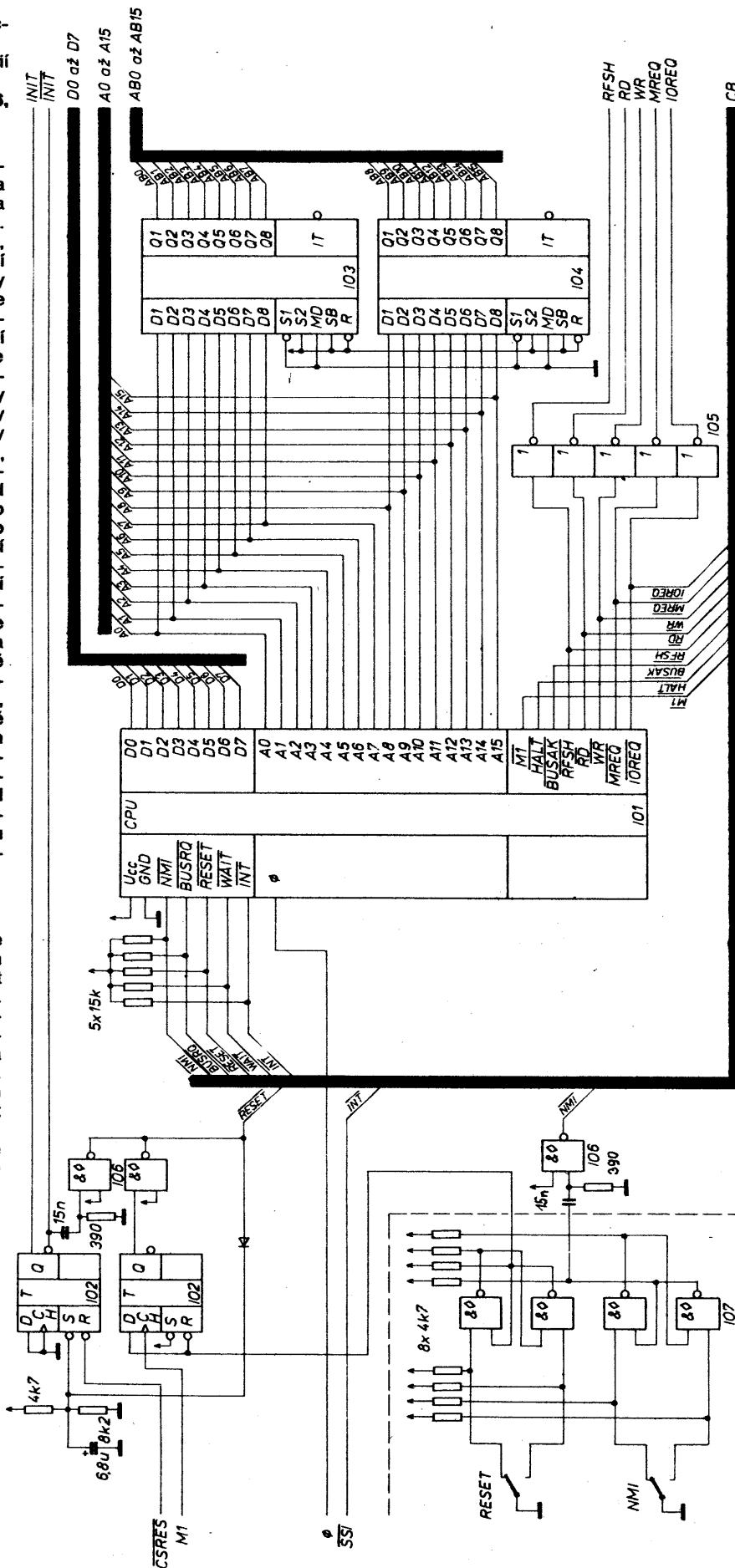
Obvodové řešení mikropočítače můžeme rozdělit do několika bloků, jejichž činnost i vzájemná spolupráce bude dále popsána:

- procesor, inicializační obvody a zesilovače adresované sběrnice — obr. 1,
- čítač adresy displeje a multiplexery adresy dynamické paměti RAM — obr. 2,
- dynamická paměť RAM, stínová paměť PROM, vyrovnávací registry dynamické paměti RAM a multiplexery signálů pro video — obr. 3,

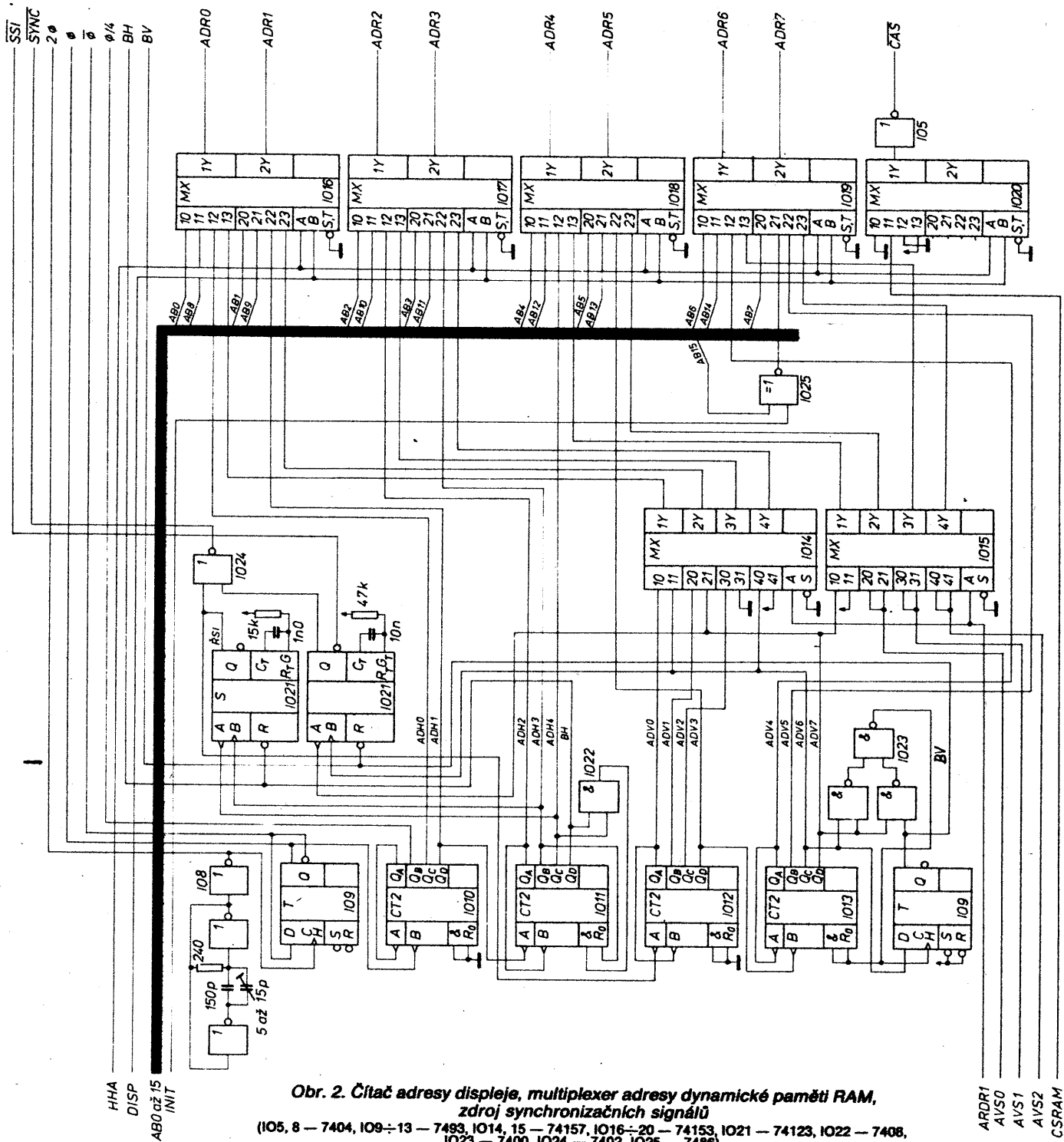
- d) obvody řízení přístupu do dynamické paměti RAM — obr. 4,
e) obvody připojení základních periférií — obr. 5,
f) obvody připojení TV a MGF — obr. 6.

ad a) Procesor je připojen svou adresovou sběrnicí (vývody A0 až A15) na systémový konektor a zároveň i na vstupy zesilovačů adresové sběrnice. Jsou realizovány dvěma obvody 3212, lze však použít libovolné neinverující zesilovače s nízkopříkonovými vstupy (např. 8282, 8286). Datová sběrnice procesoru je připojena opět na systémový konektor a vede na další bloky. Řídící sběrnice procesoru je opět celá vyvedena na systémový konektor. Vstupní signály jsou ošetřeny připojením na +5V přes odpory 15k Ω . Výstupní signály jsou zesíleny invertory TTL (RFSHN, RDN, WRN, MREQN, IDREQN) a vedeny ke zpracování do dalších bloků. Signál RESETN je generován po připojení napájecího napětí, po vynulování inicializačního klopného obvodu výstupní instrukcí a nebo po zmáčknutí přepínacího tlačítka RESET. Tento signál však musí přijít v okamžiku, kdy procesor nekomunikuje s pamětí. Po jeho příchodu se totiž okamžitě začnou měnit údaje na datové i adresové sběrnici a do právě aktivované buňky se může zapsat mylná informace. Častější je však případ, že se přepíše několik buněk, vzájemně se lišících pouze v několika adresových bitech. Tato situace může nastat prakticky pouze při ručním aktivování signálu RESET. Z toho důvodu je signál v tomto případě vytvořen teprve po příchodu prvního nastupné hrany signálu M1, čímž je zajištěna jeho správná časová koincidence s aktivací paměti.

ad b) Základní součástí tohoto bloku je oscilátor 7 MHz, který může být osazen krystalem. Na tento kmitočet může být i vydělen kmitočet jiného oscilátoru. Jak bylo ale ověřeno praktickým provozem, svou stabilitou plně postačí multivibrátor z invertorů TTL, doladěný na kmitočet 7 MHz kapacitním trimrem. Výstupní kmitočet oscilátoru je vydělen klopným obvodem D dvěma a použit jako hodinový kmitočet procesoru. Tento signál je také veden do čítače horizontální adresy znaku, tvořeného dvěma obvody 7493 a jedním hradlem AND. Tento čítač dělí v poměru 1:224, přičemž na jeho nejvyšším výstupu je střída signálu 6:8. Jako adresové výstupy se užívá šest významnějších bitů. Nejvyšší bit je označen BH (border horizontálně). Čítač adresy vertikálně je sestaven ze dvou čítačů 7493 a klopného obvodu typu D. Je inkrementován signálem RSI (řádkový synchronizační impuls) z důvodu časového zpoždění průchodu signálu čítači. Zpoždění signálu na čítači adresy může být maximálně 140 ns (po dobu, v níž má signál hodin procesoru úroveň log. 0). Paměť je totiž aktivována vždy v době, kdy jsou hodiny procesoru v úrovni log. 1. Pokud by totiž uvažované zpoždění bylo větší než 140 ns, nemohlo by se použít uvažované řešení časového multiplexu. Při změně adresy během vyčítání informace z paměti pro

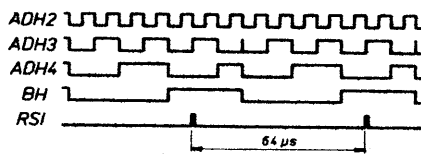


Obr. 1. Procesor, inicializační obvody a zesilovače adresové sběrnice
(IO1-Z80ACPU, IO2 — 7474, IO3, 4 — 3212, IO5 — 7404, IO6, 7 — 7403)

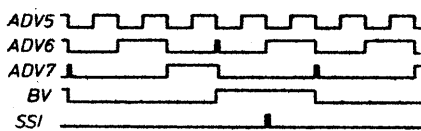


displej by tato informace nebyla jednoznačná. Vyšší bity čítačů adresy displeje musí být proto inkrementovány v době, kdy tato jednoznačnost nevádí. Pokud je to prováděno při zobrazování borderu, nevádí případně „chyby“, protože se na obraze neprojeví. Čítač vertikální adresy dělí v poměru 1:320, na jeho nejvyšším výstupu (klopný obvod) je střída 8:12. Tento signál je označen BV (border vertikálně).

Signály synchronizační směsi jsou vytvářeny dvěma monostabilními klopnými obvody (dvě poloviny obvodu 74123). Signály BH a BV jsou obvody trvale nulovány během aktivní doby zobrazování videoRAM. Prostřednictvím signálů ADH3 a ADH4, popř. ADV6 a ADV7 jsou signály synchronizační směsi generovány přibližně v 1/3 aktivní doby signálu BH a 1/2BV. Časové



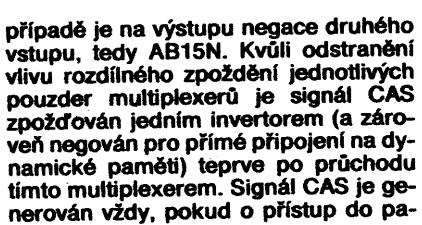
Obr. 2a. Časový diagram signálů BH a RSI



Obr. 2b. Časový diagram signálů BV a SSI

rozložení signálů BH a RSI je na obr. 2a, časový diagram signálů BV a SSI je na obr. 2b.

Pro přepínání adres dynamické paměti RAM a pro vytvoření signálu CAS, zpožděného za signálem RAS, je použito celkem pět pouzder dvojic čtyřnásobných multiplexerů. Čtyři z nich jsou použity pro přepínání nižší a vyšší poloviny adresy dynamické paměti RAM, jedna polovina pátého je použita pro generování signálu CAS. Na tuto pozici vyberte nejpomalejší z použitých multiplexerů. Adresový bit AB15 není na multiplexer veden přímo, ale přes hradlo EXOR, které je využito jako řízný invertor. Na jeho druhý vstup je připojen signál INIT a pokud je v log.0, výstup sleduje logickou úroveň druhého vstupu, tedy AB15. V opačném



```
CAS=RASD + CSRAM
kde
CSRAM=SP1.RFSHN
(INITN+AB15)
```

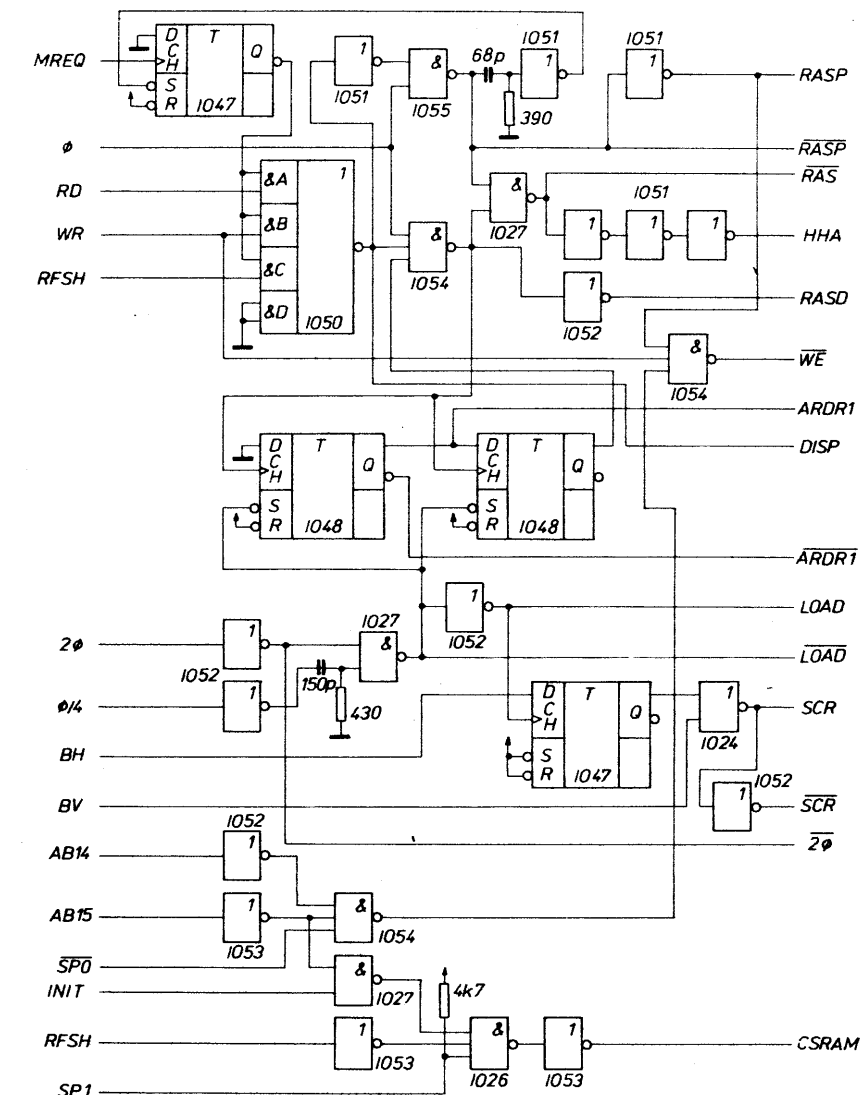
8 **mikroelektronika '88**

procesor komunikuje s vnitřní pamětí nebo s vnější pamětí. Pokud je v úrovni log.0, komunikuje s vnější pamětí, v opačném případě je aktivována paměť vnitřní. Signál SP1 je vyveden na konektor mikropočítače a přes odpor připojen na +5V. Výstupem s otevřeným kolektorem je možno jej ovládat z připojené expanzní desky paměti, a tak přepínat paměť programově. VideoRAM je umístěna stále ve vnitřní paměti, nelze ji přemístit do paměti expanzní. Celou přídavnou paměť tak lze použít pro programové účely, popř. jako RAMdisk.

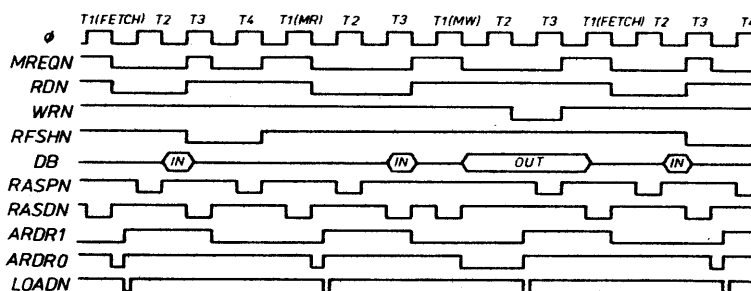
ad c) Vstupy všech pouzder dynamic-
kých pamětí RAM (kromě datových)
jsou zapojeny paralelně. Adresové
vstupy jsou připojeny na výstupy multi-
plexerů adresy dynamické paměti (sig-
nály ADRO až AD7), vstupy RASN,
CASN a WEN jsou připojeny na signály
označené stejně. Datové vstupy jsou
připojeny přímo na datovou sběrnici (D0
až D7). Datové výstupy jsou připojeny
na vstupy tří osmibitových registrů
s nízkopříkonovými vstupy (3x MH
3212). IO36 je použit jako vyrovnávací
registr pro komunikaci procesoru
s pamětí v cyklu čtení informace
z paměti. Informace je do něj zapiso-
vána signálem RASP, jeho výstup je na
datovou sběrnici připojován signálem
MREQ. RD. CSRAM, tzn. pouze
v případě, že procesor čte z vnitřní
paměti RAM. Obvod IO37 pracuje jako
vyrovnávací registr jasu, obvod IO38
jako vyrovnávací registr atributu. Do
vyrovnávacího registru jasu se informa-
ce zapisuje při ARDR1. RASD, do
vyrovnávacího registru atributu sig-
nálem ARDR1N. RASD. Informace
z těchto registrů je synchronně přepi-
sována do registrů displeje signálem
LOADN, popř. LOAD. Registr jasu je
tvořen osmibitovým posuvným regis-
trem 74165, registr atributu dvěma
pouzdry 7475. Výstup posuvného regis-
tru ovládá výstupní multiplexery jasové
informace displeje v závislosti na na-
stavení nejvyššího bitu registru atribu-
tu. Je-li tento bit (označovaný firemní
literaturou jako příznak „FLASH“) v
úrovni log.0, je při výstupu posuvného
registru v log.0 přepnut multiplexer na
jas papíru, v opačném případě na jas
inkoustu. Pokud má příznak „FLASH“
hodnotu log.1, je periodicky střídán jas
papíru s jasnem inkoustu a tak je
zabezpečeno blikání v závislosti na
výstupu obvodu IO44 (časovač 555)
kmitočtem asi 2 Hz. Právě popsaným
způsobem se vytváří obrazová informa-
ce za podmínek SCRN = log.0, tzn. je
zobrazována aktivní část informace.
V opačném případě je zobrazován jas
borderu přepnutím multiplexerů disple-
je signálem SCRN.

Stínová paměť ROM je tvořena
dvěma pouzdry paměti PROM 74S287.
Obsahuje program, který po spuštění
nastaví registr videoRAM do oblasti,
kde ji předpokládá ZX Spectrum.

Poté se volá podprogram, pro-
vádějící nahrání bloku o délce 16 kB od
adresy 8000H a po jeho úspěšném
nahrání pomocí výstupní operace
OUT (1), A „shození“ inicializačního
klopného obvodu a tím vygenerování
signálu RESET. Zároveň jsou vý-
měněny obě poloviny paměti v adreso-
vém prostoru mikroprocesoru a po
skončení RESET začne program pro-
bíhat od první instrukce nahraného
bloku dlouhého 16 kB. V případě neú-



Obr. 4. Obvody řízení přístupu do dynamické paměti RAM
(IO27, 55 — 7400, IO24 — 7402, IO51 — 53 — 7404, IO26, 54 — 7410, IO50 — 7454,
IO47, 48 — 7474)



Obr. 4a. Časový diagram činnosti řídicích obvodů

spěšné nahrávky je provedena instruk-
ce OUT (2), A. Vygenerovaný zápisový
signál lze použít po prodloužení mono-
stabilním klopným obvodem k upozor-
nění obsluhy. Stínová paměť ROM je
vybírána za podmínky splnění vztahu
INIT. AB15N. MREQ. RD (3)

ad d) Tyto obvody tvoří vlastně řídicí
část obvodů, nahrazujících obvod ULA.
Základem jsou obvody, zajišťující časo-
vý multiplex přístupu do paměti a řízení
vyrovnávacích registrů výstupu paměti.
Pochopení jejich činnosti je nejsnazší
z časového diagramu na obr. 4 a.

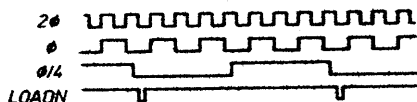
Signál MREQ začíná vždy se sestupnou
hranou hodinového signálu procesoru,
má však různou délku. V případě
obnovovacího cyklu trvá pouze jednu
periodu, v případě čtení kódu instrukce
1,5 periody a v případě čtení či zápi-
su dat do paměti 2 periody.
K jednoznačnému určení časového
okamžiku přístupu procesoru do pa-
měti je použit pomocný klopný obvod
D, jenž je nastavován nástupnou hra-
nou signálu MREQ. Nulován je vzestup-

nou hranou signálu RASPN, čímž je zajištěno generování právě jednoho signálu RASP na každou žádost procesoru o přístup do paměti. Výstupní signál tohoto klopného obvodu označme QMREQ. Signál RASP je pak generován podle rovnice:

$$\text{RASP} = \text{QMREQ} \cdot (\text{RD} + \text{WR} + \text{RFSH}) \quad (4)$$

Pokud je tedy paměť aktivována procesorem pro čtení, je informace čtena z vyrovnávacího registru, protože v době, kdy procesor čeká na data na datové sběrnici, není již paměť procesorem aktivována. Při zápisu obvodu naopak čekají na příchod signálu WR, aby nebyla paměť zbytečně dlouho blokována. Obnovovací cyklus proběhne bez časového posunu, jen je na polovinu zkrácen. Je tedy zajištěno, že libovolná komunikace s pamětí bude trvat právě po dobu trvání signálu Φ v úrovni log. 1. Paměť je aktivována nejčastěji každou druhou periodu hodinového kmitočtu, zbývá tedy dost času pro komunikaci obvodů displeje s pamětí.

Základním signálem, ovládajícím časování obvodů displeje, je signál přepisu informace do registrů displeje LOAD. Je generován ze signálu 2Φ a $\Phi/4$ tak, aby přicházel v první půlperiodě hodinového signálu ($\Phi = \log.0$) po sestupné hraně $\Phi/4$, začínal se sestupnou hranou 2Φ a trval asi 40 ns. Zpoždění je nutné, aby se ustálila informace ve výstupech čítačů horizontální adresy znaku, synchronizace se signálem 2Φ zabezpečuje plynulý přechod mezi znaky a délka trvání je pouze nezbytná pro reakci ostatních obvodů. Jeho časový průběh s ostatními signály je ukázán na obr. 4b.



Obr. 4b. Časové umístění signálu LOADN

Signálem LOADN je nastaven dvoubitový posuvný registr, tvořený dvěma klopnými obvody D (IO48) do stavu 11B. Jeho vstup je trvale připojen na log.0, informace v něm je posouvána vzestupnou hranou signálu RASDN, tj. po každém vyčtení bajtu pro displej. Vyšší bit udává adresu vyrovnávacího registru paměti, do kterého se bude zapisovat a zároveň ovládá multiplexery, přepínající adresu jasu na adresu atributu (IO14, IO15). Nižší bit udává, zda je ještě nutno aktivovat vyčítání z paměti pro displej. Po příchodu LOAD se nejprve vyčte jas, poté atribut. Vyšší bit je označen ARDR1, nižší ARDR0. Signál RASD, udávající, že s pamětí komunikují obvody displeje, je tvořen podle rovnice:

$$\text{RASD} = \text{NON} (\text{QMREQ} \cdot (\text{RD} + \text{WR} + \text{RFSH})) \cdot \text{ARDR0}$$

Signál RASN, ovládající přímo odpovídající vstupy dynamických pamětí RAM, je generován následovně:

$$\text{RAS} = \text{RASP} + \text{RASD} \quad (6)$$

Signál DISP určuje s předstihem půl periody hodinového signálu, zda se bude komunikovat s pamětí kvůli požadavku procesoru (log.0) nebo zda o přístup žádá displej. Je dán vztahem:

$$\text{DISP} = \text{NON} (\text{QMREQ} \cdot (\text{RD} + \text{WR} + \text{RFSH}))$$

Pro přepínání multiplexerů adresy dynamické paměti RAM je využit signál HHA, který dostaneme zpožděním signálu RASN průchodem třemi invertoři. Zpoždění, vzniklé na těchto invertořích, spolu se zpožděním signálů na multiplexerech, zcela vyhovuje potřebám dynamických pamětí a není třeba jej vyrábět pomocí pasivních článků RC, s jejichž nastavováním bývají potíže.

Dále je třeba mít k dispozici signál, určující, zda se v daném okamžiku bude zobrazovat jasová informace vytvořená obvody displeje nebo jas borderu. K tomuto účelu je využit signál SCR:

$$\text{SCR} = \text{NON} (\text{BV} + \text{QBH}) \quad (8)$$

kde QBH označuje výstup klopného obvodu D, na jehož datový vstup je přiveden signál BH a je do něj přepisován vzestupnou hranou signálu LOAD. Tento klopný obvod musí totiž kompenzovat zpoždění signálů, vzniklé na vyrovnávacích registrech paměti.

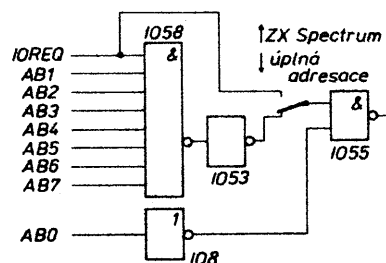
Pro ovládání dynamické paměti RAM je nutný ještě signál WEN. Při jeho vytvoření musíme mít na paměti, zda můžeme povolit zápis do části paměti, kde je u normálního Spectra umístěna paměť ROM, či nikoliv. Je vhodné mít možnost toto programově ovládat pomocí signálu, který si označíme SPO:

$$\text{WE} = \text{RASP} \cdot \text{WR} (\text{AB14N} + \text{AB15N} + \text{SPO}) \quad (9)$$

Signál SPO můžeme totiž připojit na výstup pomocného klopného obvodu nebo na systémový konektor a programově povolovat možnost zápisu do nejnižších 16kB. V tom případě je však nutno zakázat zápis ještě před odepnutím stínové paměti ROM. Za podmínky $\text{SPO} = \log.0$ je zápis zakázán. Na obr. 4 jsou ještě obvody, zajišťující tvorbu signálu CSRAM podle rovnice (2.)

ad e) Každý mikropočítač je třeba nějak ovládat. K tomu slouží základní periferie — klávesnice, zobrazovací jednotka (v našem případě TV displej), zvukový měnič pro upozornění obsluhy a základní vnější paměť pro uchování dat (magnetofon). Klávesnice je připojena obdobným způsobem jako u Spectra — horních osm bitů posílené adresové sběrnice je přes diody zapojeno na osm vodorovných řad klávesnice (každá řada je rozdělena v polovině na dvě). Pět svislých sloupců (každý sloupec má osm tlačítek) je zapojeno na spodních pět bitů vstupního portu s adresou OFEH. Podrobně je to vidět na obr. 5. Protože port OFEH je

obousměrný a potřebuje mít možnost zapamatovat pět bitů ve výstupním směru, je použito dvou obvodů 3216, jednoho čtyřbitového registru 7475 a jednoho klopného obvodu D.



Obr. 5a. Úprava adresace

TV displej je ovládán z podstatné části signály, vytvářenými podle obr. 3. Z portu OFEH je k nim přidána pouze informace o barvě borderu (signály BRB, BRR a BRG). Komunikace s magnetofonem je realizována po dvou vodičích. Signál pro nahrávání je poslán po vodiči označeném MGFIN a čten z vodiče MGFOUT. Reprodukční je ovládán signálem REPRO. Pomocí základní logiky jsou vydekódovány ovládací signály pro port OFEH takto:

$$\text{CSFEH} = \text{IOREQ} \cdot \text{AB0N} \cdot \text{AB1} \cdot \text{AB2} \cdot \text{AB3} \cdot \text{AB4} \cdot \text{AB5} \cdot \text{AB6} \cdot \text{AB7} \quad (10)$$

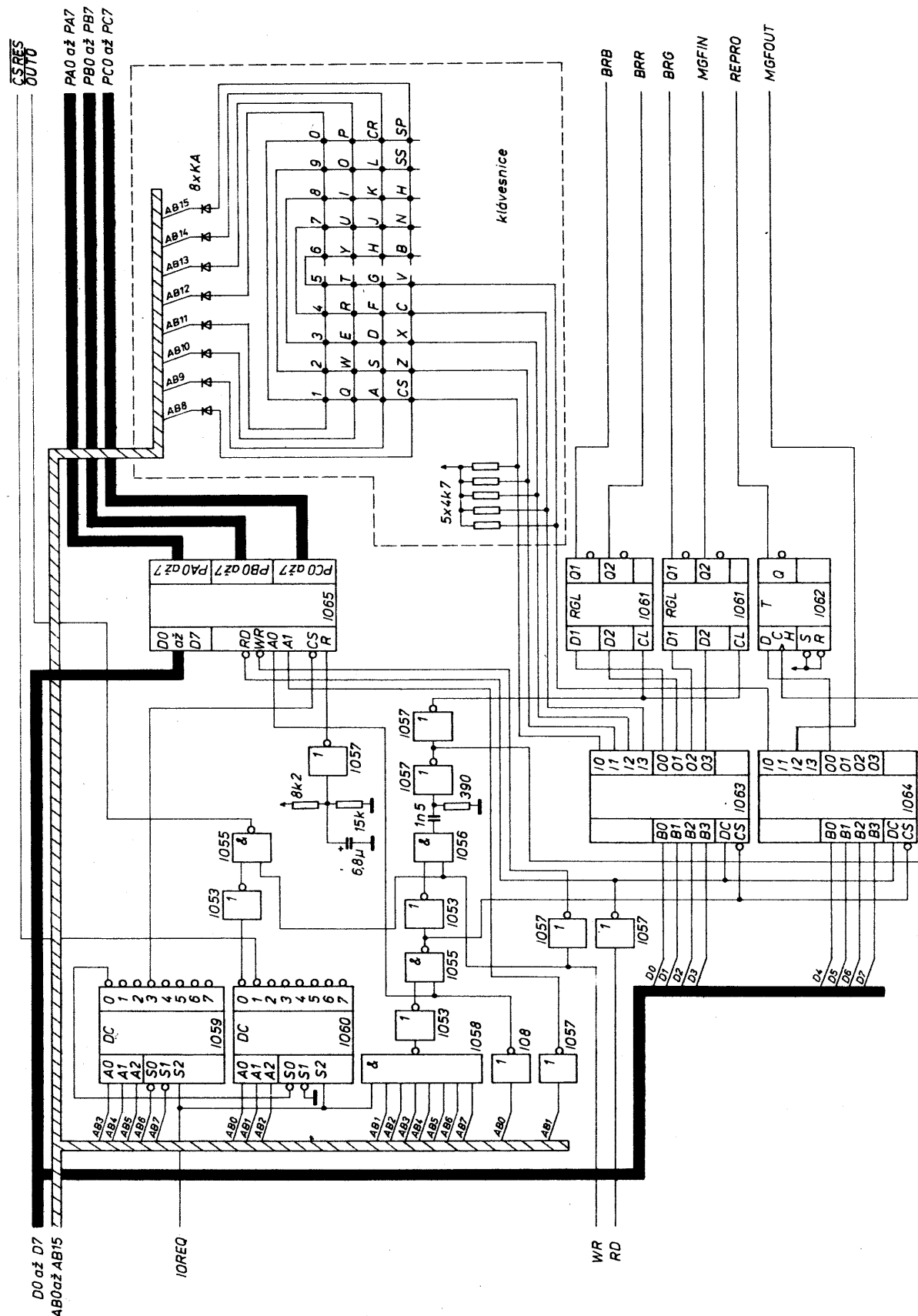
Protože originální Spectrum používá k adresování tohoto portu pouze AB0, pro případnou plnou kompatibilitu je nutno výběrový signál vytvořit následovně:

$$\text{CSFEH} = \text{IOREQ} \cdot \text{AB0N}$$

Lze to jednoduše udělat vynecháním IO54 a následného invertoru, nebo za tento invertor zapojit přepínač, kterým by se místo pevného spojení tohoto invertoru s vstupem hradla NAND mohlo volit mezi výběrovým signálem podle (10) a (11). Úprava je znázorněna na obr. 5a. Dále na obr. 5 je nakreslen obvod 8255 (IO65), který slouží pro připojení tiskárny a ovládače. Protože obsahuje tři využitelné programovatelné vstup-výstupní brány, lze jej použít samozřejmě i k dalším aplikacím. Jeho brány jsou do adresového prostoru zařazeny takto:

brána:	adresa:
CW (řídící)	1CH (28D)
A	1FH (31D)
B	1EH (30D)
C	1DH (29D)

Je vidět, že adresa portu A je stejná jako adresa vstupního portu pro ovládač Kempston. Protože po zapnutí napájecího napětí je obvod 8255 naprogramován do vstupního módu pro všechny brány, nemusí se ani přeprogramovávat, pokud chceme použít program pro tento typ ovládače. Je vhodné upozornit, že tento obvod není součástí originálního Spectra, ale prakticky každý uživatel dříve či později narazí na nutnost připojení nějaké periferie k mikropočítači. Z tohoto důvodu byl tento obvod zahrnut přímo do základní sestavy navrhovaného mikropočítače. Porty jsou vyvedeny na uživatelský konektor.

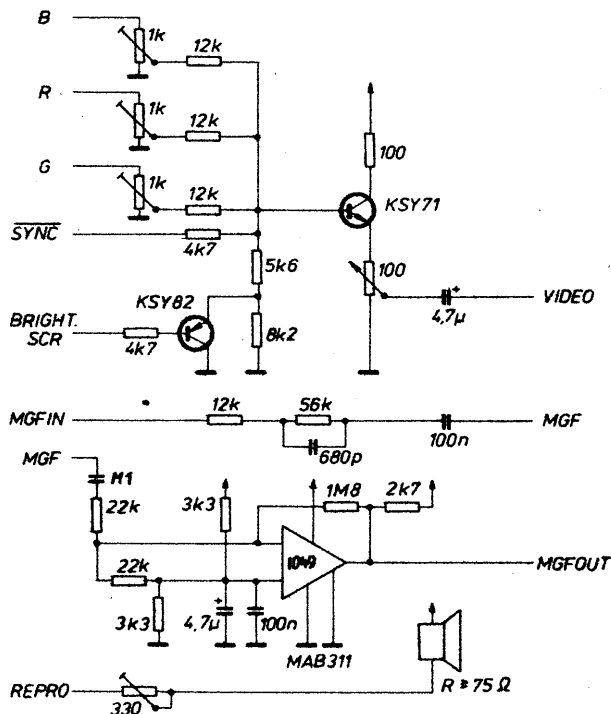


Obr. 5. Připojení periférií
(IO55 — 7400, IO58, 53, 57 — 7404, IO56 — 7408, IO62 — 7474, IO61 — 7475, IO58 —
— 7430, IO59, 60 — 3205, IO63, 64 — 3216, IO65 — 8255)

ad f) Po několika vyzkoušených variantách obvodů připojujících TV displej a magnetofon byly realizovány obvody, jejichž schéma je na obr. 6. Jsou z mnoha typů autorem vyzkoušených zapojení nejjednodušší bez snížení kvality a zejména s nejnižšími nároky na seřizování.

Výstup na magnetofon má mírné zdůraznění vyšších kmitočtů při nahrávání. Toto jednoduché opatření přineslo překvapivě zlepšení spolehlivosti čtení záznamu na jiných magnetofonech, než byl záznam pořízen. Vstupní komparátor je realizován jedním integrovaným komparátorem, který má

zavedenu malou hysterii. Za předpokladu kvalitního vstupního signálu pracuje spolehlivě již od mezivrcholové úrovně signálu 200 mV. Protože vstupní



Obr. 6. Připojení televizního displeje a magnetofonu

a výstupní signál nejsou spojeny, neprojevuje se vazba mezi nimi (nedochází k ní). Můžeme tedy magnetofon provozovat s trvale zapojenými oběma kabely a nebo jej spojit přímo pětikolovou nahrávací šňůrou (pozor, u některých magnetofonů zahraniční výroby s jinak zapojenými výstupy než udává naše norma nemusí toto propojení fungovat, tehdy je nutno použít sluchátkový výstup).

TV displej je připojen přes dva tranzistory a součtovou odporovou matici. Jasový signál je vytvořen sečtením signálů R, G a B, což jsou barevné složky, signálu přijetí BRIGHT, SCR a signálu synchronismu SYNCN. Signál přijetí je použit v součinu se signálem udávajícím aktivní část obrazovky z toho důvodu, aby nemohlo dojít k přijetí borderu. Jednotlivé barevné složky signálu jsou vedeny z výstupů multiplexeru na odporové trimry, kterými se nastaví jasový příspěvek jednotlivých barev. Signál synchronismu je přičítán přímo bez odporového děliče. Signál přijetí je přičítán přes tranzistor p-n-p. Jinak by se totiž těžko dosáhlo zobrazení formátu Spectra, protože tam signál BRIGHT zvyšuje kontrast zobrazení a úroveň černé zůstává konstantní. Sečtené signály jsou převedeny na malou impedanci emitorovým sledovačem, který má v emitoru potenciometr. Výsledný videosignál je odebrán z běžce potenciometru v pozitivní polaritě. Potenciometr byl použit, aby bylo možné vyrovnání citlivosti různých monitorů.

Reproduktor je připojen na signál REPRO vůči +5V přes potenciometr, umožňující regulaci hlasitosti. Je vhodné použít reproduktor o co největší impedanci.

TABULKA č. 1

SYSTÉMOVÝ KONEKTOR	
FRB 62 KLÍČ: E3	
1 A0	2 A1
3 A2	4 A3
5 A4	6 A5
7 AG	8 A7
9 A8	10 A9
11 A10	12 A11
13 A12	14 A13
15 A14	16 A15
17 D0	18 D1
19 D2	20 D3
21 D4	22 D5
23 D6	24 D7
25 UCC	26 GND
27 IORQN	28 RDN
29 MREQN	30 WRN
31 HATLN	32 RFSHN
33 MIN	34 BUSAKN
35 NMN	36 BUSRQN
37 INTN	38 RESETN
39 WAITN	40 0
41 AB0	42 SP1
43 AB2	44 AB1
45 CS(10-17)N	46 CS(20-27)N
47 CS(28-2F)N	48 CS(30-37)N
49 CS(38-3F)N	50 CS3N
51 CS4N	52 CS5N
53 CS6N	54 CS7N
55 DB0	56 DB1
57 DB2	58 DB3
59 DB4	60 DB5
61 DB6	62 DB7

UŽIVATELSKÝ KONEKTOR	
FRB 30 KLÍČ: E5	
1 UCC	2 GND
3 PA7	4 PAG
5 PAS	6 PA4
7 PA3	8 PA2
9 PA1	10 PA0
11 PC7	12 PC6
13 PC5	14 PC4
15 UCC	16 GND
17 PC3	18 PC2
19 PC1	20 PC0
21 PB7	22 PB6
23 PB5	24 PB4
25 PB3	26 PB2
27 PB1	28 PBQ
29 UCC	30 GND

4. MECHANICKÉ PROVEDENÍ

Prototyp byl postaven na dvou univerzálních deskách s plošnými spoji rozměru „Velká eurokarta“, sešroubovaných nad sebe. Na spodní desce je umístěn kompletní mikropočítač, na horní je tlačítková klávesnice, obvody připojení MGF, TV, reproduktor a mikropočítače RESET a NMI. Jsou zde také konektory připojení napájecího zdroje, monitoru a magnetofonu. V případě dalších dvou kusů, které byly sestaveny, byly obvody 3212 nahrazeny obvody 8282 a do získaného prostoru bylo umístěno připojení TV a MGF. Obě desky jsou propojeny 30-ti vývodovým konektorem FRB, postaveným „nastojato“. Systémový konektor je FRB62, uživatelský FRB 30, oba dva jsou na základní desce. Zapojení jejich vývodů je v tabulce 1.

Při umístění desek v malé vzdálenosti nad sebou se součástky na spodní desce příliš zahřívají. V průběhu 2 až 3 hodin dojde k přehřátí a tím i k havárii systému. Je nutné zajistit nucené chlazení vzduchem, proudícím mezi deskami. K boku spojených desek je připevněn amatérsky zhotovený větrák, vhánějící vzduch mezi ně. Motorek má z kazetového magnetofonu a válcovou vrtulku s listy z PVC folie. Je napájen přímo z počítačového rozvodu +5V. Toto chlazení stačí zcela k udržení teploty součástek v únosných mezích prakticky neomezeně dlouhou dobu.

Klávesnice je zhotovena ze 40 spínacích tlačítek WK 55929 s amatérsky zhotovenými hmatníky z Eprosinu. Mohou se použít i libovolná jiná spínací tlačítka, mikropočítače nebo membránová klávesnice.

5. SOUČÁSTKY, OŽIVENÍ A UVEDENÍ DO PROVOZU

Jak již bylo řečeno v úvodu, jedinými typy součástek, které nebyly na trhu v době realizace prvního vzorku (leden 1987) byl procesor UA880D a paměti typu 4164 s dobou přístupu pod 150 ns. V prototypu byl použit procesor UB880D, který byl vyzkoušen na hodinovém kmitočtu 3,5 MHz a paměti NEC D4164C-12 se zaručenou dobou přístupu 120 ns. Protože není v silách běžně vybaveného amatéra určit mezní kmitočet procesoru, případný zájemce by měl použít procesor se zaručeným mezním kmitočtem vyšším než 3,5 MHz. V úvahu přicházejí typy UA880D a Z80A. Má-li někdo možnost vybrat z více kusů obvodů UB880D, je vhodné to udělat pomocí měření zpoždění signálů řídící sběrnice za hodinovým signálem a vybrat kusy s co nejmenším zpožděním. Některé série (značeno zespodu obvodu) bývají vhodné prakticky celé, jiné jsou zase skoro celé pomalejší.

Doba přístupu u paměti je výrobcem značena v desítkách ns za pomlčkou, udávající typ obvodu. U konkrétního obvodu je ovlivněna především teplotou čipu. Na tu má vliv zatížení výstupů a vstupní signály. Pokud budou dodrženy následující rady, je pravděpodobné, že v počítači vyhoví i paměti s dobou přístupu 150 ns. Autor vyzkoušel celkem čtyři čtveřice takovýchto

paměti od tří výrobců a všechny fungovaly bezchybně. Především je nutno co nejméně zatěžovat datové výstupy paměti — v konstrukci jsou připojeny na každý výstup tři vstupy obvodu 3212. Přestože někteří výrobci udávají zatížitelnost až dva vstupy TTL, není vhodné paměti provozovat takto zatížené. Dále je vhodné vstupy nepřipojovat přímo, ale přes sériové rezistory řádu desítek ohmů (27 až 56 ohmů). Dynamická paměť je totiž vnitřní strukturou blízká analogovým obvodům a tímto opatřením dále zmenšíme její příkon.

Stavbu je vhodné začít základním oscilátorem 7 MHz, jehož kmitočet nastavíme čítačem. Dále oživíme čítače adresy displeje a obvody vytvoření synchronizačních impulsů. Pokud nyní zapojíme sčítací obvod pro video, na monitoru bychom měli dostat zasyntetizovaný obraz obdélníku. Nyní zapojíme procesor, zesilovač adresové sběrnice, inicializační obvody a obvody řízení přístupu do dynamické paměti RAM. Při nezasunutém procesoru do objímky zkontrolujeme tvorbu signálů RASD, RAS, LOAD, HHA, ARDR1 a ARDR1. Signály RAS a RASD by měly být vždy dva po signálu LOAD, pak zase po dobu dvou period by neměl být signál RAS ani RASD aktivován. Nyní zkontrolujeme tvorbu signálu LOAD. Nemáme-li k dispozici osciloskop, lze to udělat i pomocí připojeného TV displeje. Je-li délka impulsů, vytvářených derivačním článkem RC nevyhovující, na displeji nebude obdélník čistý nebo nebude vůbec. Na místo kondenzátoru 150 pF dáme kondenzátor 82 pF a jeho kapacitu postupně zvětšujeme, nakonec použijeme kondenzátor s asi o 20 až 30 pF větší kapacitou, než byla nejmenší vyhovující. Většinou to bude 120 nebo 150 pF. Je-li vše v pořádku, zasuneme procesor do objímky a znovu zkontrolujeme signály včetně signálů RASP, a WE. Zapojíme multiplexery adresy paměti a zkontrolujeme časové zpoždění CAS za RAS. Odpovídá-li povolenému zpoždění, udávanému výrobcem, postupujeme dále, jinak jej upravíme vkládáním nebo vypouštěním sudého počtu invertorů ve zpožďovacím řetězci. Toto by se ale nemělo stát. Pokud zpoždění nesouhlasí, zkontrolujeme napájecí napětí přímo na vývodech integrovaných obvodů. Zpoždění totiž závisí nejvíce právě na napájecím napětí.

Dále pokračujeme zapojením vyrovnávacích registrů paměti, registry displeje, multiplexery displeje a paměťmi RAM a PROM. Před zasunutím paměti RAM do objímek zkontrolujeme napájecí napětí (mají +5V na vývodu 8 a 0V na vývodu 16). Nakonec zapojíme obvody připojení periférií a obvod 8255. Zasuneme paměti PROM a zapojíme napájecí napětí, na obrazovce bude nějaký pravidelný pravoúhlý obrazec (pruhy, šachovnice) v černé a přijasně bílé. Polovina paměťových buněk se po zapnutí napájecího napětí totiž přepne do log.0 a polovina do log.1. Obrazec závisí na vnitřní struktuře paměti.

Po připojení magnetofonu a jeho spuštění by měl začít probíkat border, indikující nahrávání. Můžeme zkusit nahrát obsah originální paměti ROM, který jsme již dříve nahráli ze Spectra příkazem SAVE „ROM“ CODE 0,16384. Po skončení nahrávání by

měla proběhnout inicializační rutina a vypsat se úvodní hlášení.

Pokud jsme se úspěšně dostali až sem, zbývá už pouze nastavit trimry jednotlivých barevných složek jasu. Dá se to udělat pomocí pomocného programu, například takovéhoto:

```
10 FOR B= 0 TO 1
20 FOR P= 0 TO 7
30 PRINT PAPER P;BRIGHT B;";
40 NEXT P
50 NEXT B
60 GO TO 10
```

Po spuštění tohoto programu se na obrazovce zobrazí 16 jasově odlišných pruhů, přičemž prvních osm je od černého po bílý nepřijasných a druhá polovina je přijasně. Pomocí odporových trimrů nastavíme stupnici šedí. V řádku 30 zkušebnímu programu změním počet mezer ze dvou na šestnáct a zkontrolujeme konstantní úroveň černé bez i při přijasnění. Pokud vizuálně nesouhlasí, nastavíme ji změnou odporů 5,6 kΩ a 8,2 kΩ. Tímto je nastavení počítače skončeno.

Závěrem bych chtěl zdůraznit, že stavba mikropočítače patří k složitým záležitostem v amatérských podmínkách a je vhodná pouze pro vyspělé amatéry, mající dostatečné zkušenosti s číslicovou a mikroprocesorovou technikou. Je vhodné nepodcenit takové „banálnosti“, jako se zdá rozvod a blokování napájecího napětí. Na schématu to není naznačeno, ale je použito asi 40 blokovacích keramických kondenzátorů a 15 tantalových kapek 47 μF. Z vlastní trpké zkušenosti dodávám, že to není přehnané opatření. Nesmí se také podcenit průřez napájecích vodičů, protože odběr obvodů TTL je značný.

Tento mikropočítač nebyl od začátku stavěn s úmyslem postavit kopii Spectra, pokud by se ke stavbě chystal někdo s tímto úmyslem, doporučuji mu koupit si originál. Smyslem stavby bylo vytvořit mikropočítač jednoduše rozšířený se širokými možnostmi ovládání periferních zařízení a přitom využít obrovské knihovny programů na ZX Spectrum.

Výpis paměti PROM je na straně 29!!!

Použité integrované obvody

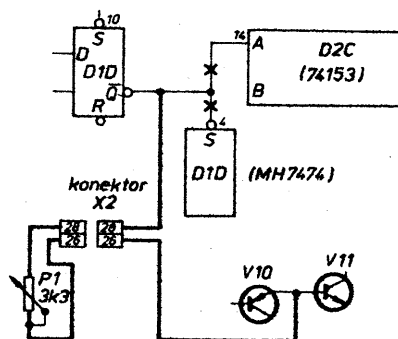
MH7400 — 3 ks (IO23,27,55)
UCY7402 — 1 ks (IO24)
MH7403 — 2 ks (IO8,7)
MH7404 — 6 ks (IO5,8,51,52,53,57)
UCY7408 — 2 ks (IO22,56)
MH7410 — 2 ks (IO26,54)
MH7430 — 1 ks (IO58)
MH7454 — 1 ks (IO50)
MH7474 — 4 ks (IO2,47,48,82)
MH7475 — 3 ks (IO40,41,81)
UCY7486 — 1 ks (IO25)
MH7493 — 5 ks (IO9,10,11,12,13)
UCY74123 — 1 ks (IO21)

UCY74153 — 7 ks
(IO16,17,18,19,20,45,46)
UCY74157 — 2 ks (IO14,15)
UCY74165 — 1 ks (IO39)
MH74S287 — 2 ks (IO42,43)
MH3205 — 2 ks (IO59,60)
MH3212 — 5 ks (IO3,4,36,37,38)
MH3216 — 2 ks (IO63,64)
MHB3255A — 1 ks (IO65)
NE555 — 1 ks (IO44)
Z80A CPU — 1 ks (IO1)
4164-12 — 8 ks
(IO28,29,30,31,32,33,34,35)
MAB311 — 1 ks (IO49)

Úprava desky AND-1 SAPI-1

Ing. Pavel Švarc

Pro použití mikropočítače SAPI-1 ke sběru a předzpracování dat je účelné používat dvojí jas zobrazovaných textů. Pro tento účel je možné upravit desku AND-1 takto:



Obr. 1. Úprava desky AND-1, tučně vyznačeny nové spoje

Přerušit spoj mezi výstupem 8 obvodu D1D a vstupy 14 obvodu D2C a 4 obvodu D1D (značení podle průvodní dokumentace) a na výstup 8 obvodu D1D napojit potenciometr 3,3 kΩ (P1) typ TP052C, jehož jezdec je připojen na bázi tranzistoru V11. Potenciometr je vložen do kryty konektoru kabelu KB-06, který je součástí dodávky, bez dalšího uchycení místo pertinaxové krytky, hřídel je zkrácena o 7 mm a osazena přístrojovým knoflíkem WF24313. Na špičku 28 konektoru TX 514 30-X2 je vyveden výstup 8 (D1D) a na špičku 26 pak báze V11. Program výstupu znaku na displej OUTDIS je třeba upravit vymazáním instrukce CZ IPOZ na adrese 0C9EH (nebo vložení 3 instrukcí NOP, pokud není možno přeadresovat). Při vložení hodnoty C0H do MODE je znak zobrazen s normálním jasnem, při vložení 00H pak potlačeným jasnem, který je možno plynule regulovat potenciometrem P1. Ostatní modifikace zobrazení znaku zůstávají stejné (blikající, podtržený) pro potlačený jas. Velikost znaků je nyní jednotná — 5 x 7 bodů.

MODULOVÁ SESTAVA

PRO MIKROPOČÍTAČ ZX SPECTRUM

Ing. J. Doležal

Příspěvek je určen pro ty zájemce o mikropočítačovou techniku a zejména majitele mikropočítače Sinclair ZX-Spectrum, kteří chtějí proniknout do tajů měřicí a regulační techniky pomocí mikropočítačů.

Proto jsem navrhl, realizoval a odzkoušel následující moduly-jednotky, které je schopen při pečlivé práci postavit i začátečník.

K jejich ovládání není třeba rozsáhlých znalostí strojového kódu mikropočítače Z80. V první fázi pokusů bohatě vystačí znalosti jazyka BASIC. Po odzkoušení modulů a získání prvních praktických zkušeností můžeme přistoupit i k pronikání tajemství programování ve strojovém kódu a tím vlastní komunikaci s periferním zařízením urychlit. Modulová koncepce umožňuje k modulu portů připojovat další libovolné jednotky.

1. Modul PIO

Tento modul obsahuje programovatelný periferní obvod MHB8255, který vytváří stykové rozhraní mezi mikropočítačem a dalšími obvody. Zapojení, které bylo zveřejněno v (1), se ukázalo vhodné i pro tento případ a bylo použito v podstatě beze změny. Proto nebudu uvádět popis zapojení ani tabulku nastavení kanálů.

Pro toto zapojení byl ale navržen nový oboustranný plošný spoj o rozměrech 107,5 x 60 mm, který je rozměrově shodný s ostatními moduly. Tento modul se připojuje ke sběrnici mikropočítače přes konektor WK 46580, nebo přes konektor FRB TY 5176211 a redukci podle (2). Oba integrované obvody jsou zasunuty do objímek. Výstupní konektor pro připojování dalších modulů je typu FRB TX 5183113.

Schéma zapojení je na obr. 1, rozmístění součástek na desce s plošnými spoji na obr. 2, obrazec plošných spojů je na obr. 3 a 4.

Po pečlivé prohlídce plošných spojů a případném proměření ohmetrem zasuneme do objímek integrované obvody, modul připojíme na konektor sběrnice a zapojíme počítač. Počítač by měl pracovat normálně. Nyní nastavíme PIO pomocí registru CWR do módu 0 a do výstupního režimu příkazem

OUT 127, 128.

Pak zadáme **1 INPUT X: PRINT X**
2 OUT 31: GOTO 1
RUN

Zadáme střídavě 0 a 255 a na vývodech 41 až 55 by se měla střídavě objevovat log. 0 a log. 1. Tímto způsobem ověříme správnou funkci kanálu B (adresa 63 a vývody 1 až 15) a C (adresa 95 a vývody 27 až 35).

Obr. 1. Schéma zapojení PIO

Seznam součástek

Polovodiče:

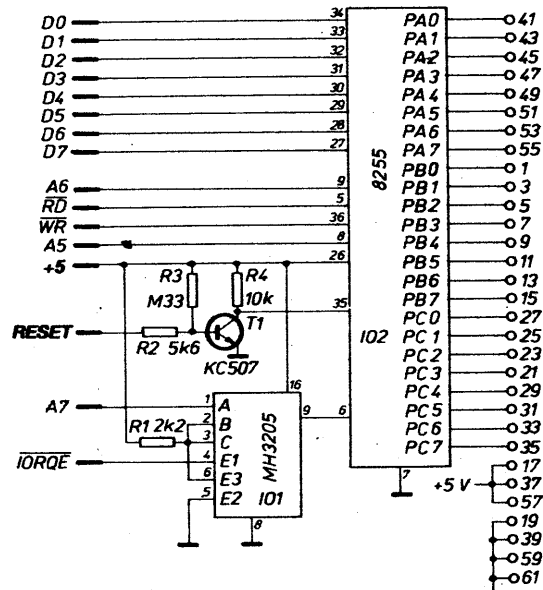
IO1 MH3205
IO2 MHB8255
T1 KC507

Rezistory:

R1 2,2 kΩ, TR 112
R2 56 kΩ, TR 112
R3 330 kΩ, TR 112
R4 10 kΩ, TR 112

Konstrukční části:

oboustranný plošný spoj
konektor WK465 80
konektor FRB TX 518 3113
objímka 2x 8 vývodů 6AF497679
objímka 2x 20 vývodů TX 7875401

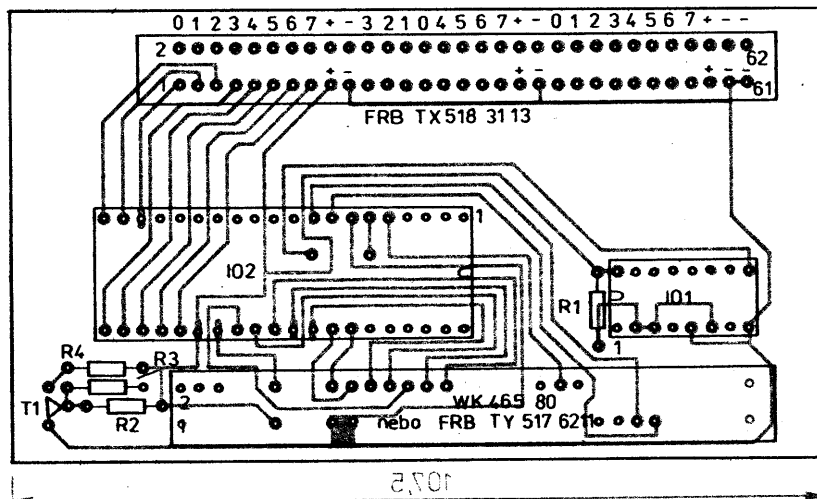


Tab. 1.

adresování obvodu MHB 8255	
řídící registr CWR	127
kanál A	31
kanál B	63
kanál C	95

Nyní nastavíme PIO do vstupního režimu příkazem OUT127, 155 a zadáme

10 LET X = IN 31 : PRINT X
11 POKE 23692, 100 : GOTO 10
RUN 10



Obr. 2. Rozmístění součástek na desce s plošnými spoji modulu PIO W501

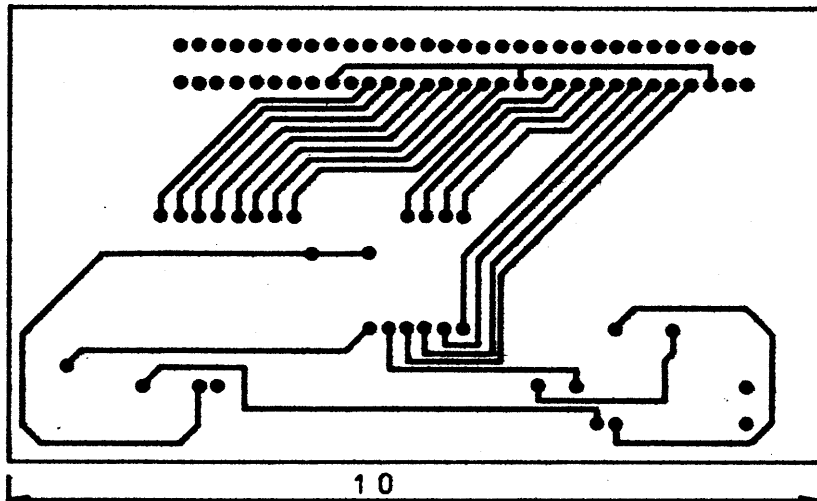
Na obrazovce se objeví sloupec nul. Budeme-li na vývody 41 až 55 střídavě přivádět log. 0 a log. 1, vždy se nám objeví na obrazovce číslo odpovídající této binární kombinaci. Stejně vyzkoušíme i kanály B a C.

2. MODUL PŘEVODNÍKŮ A/D a D/A

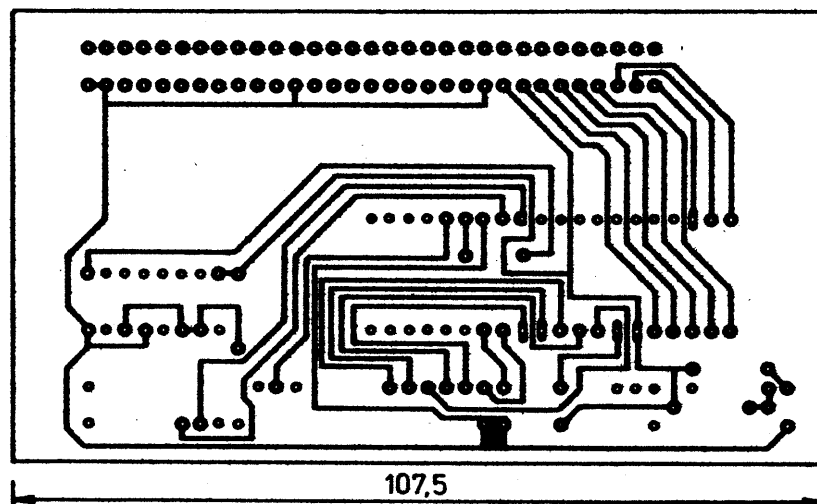
Na této desce je umístěn osmibitový aproximační převodník A/D a osmibitový převodník D/A. Stykový obvod MHB8255 s těmito převodníky komunikuje v základním módu 0 a to převodník A/D kanálem A a převodník D/A kanálem B. Kanál C zůstává nevyužit.

Popis zapojení

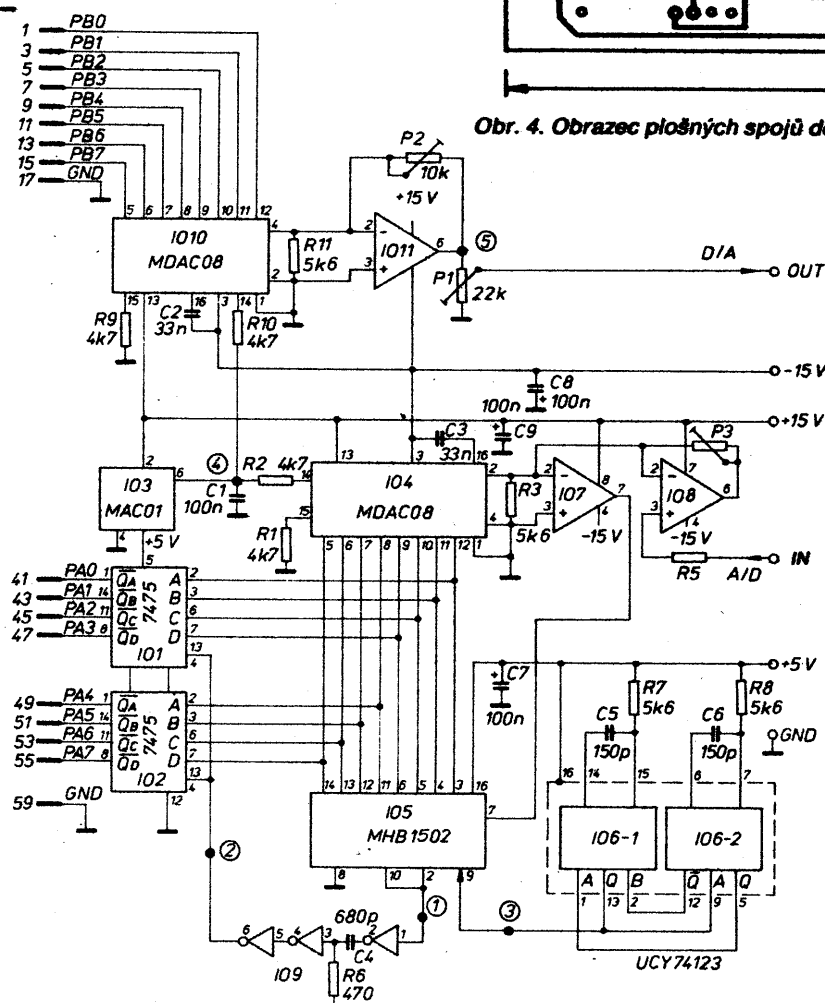
Převodník A/D, použitý v tomto zapojení, vychází ze zapojení zveřejněného v [4]. Schéma zapojení obou převodníků je na obr. 5. Převodník je tvořen osmibitovým aproximačním registrem MHB1502 (IO5), který generuje všechny řídící signály převodníku A/D s postupnou aproximací. Registr je řízen hodinovými impulsy z dvojice monostabilních klopných obvodů UCY 74123 (IO2). Ve funkci převodníku D/A je užít integrovaný obvod MDAC08 (IO4). Tento obvod má zaručenou diferenciální nelinearitu ± 1 LSB a dobu ustálení 85 až 150 ns.



Obr. 3. Obrazec plošných spojů desky W501 modulu PIO — strana součástek



Obr. 4. Obrazec plošných spojů desky W501 modulu PIO — strana bez součástek



Pro komparaci vstupního proudu převodníku A/D a výstupního proudu převodníku D/A je užít komparátor MAC111 (IO7) se zesílením $2 \cdot 10^5$ a dobou zpoždění 200 ns. Referenční zdroj napětí +10 V je osazen stabilizátorem MAC01 (IO3). Převodník A/D v tomto zapojení pracuje v asynchronním režimu. To znamená, že hodinový kmitočet 1300 kHz je přiveden na vstup CL IO5.

Po připojení převodníku na napájecí napětí se na vývodu CC objeví log. 0, která je přivedena na vstup S a uvede aproximační registr do výchozího stavu. Vzestupná hrana hodinového impulsu ho pak spustí. V převodníku D/A se výchozí číslicový signál převede na odpovídající proud, který se pak komparuje v komparátoru MAC111 (IO7) se vstupním proudem převodníku A/D, daným jeho vstupním napětím a odporem rezistoru, určujícího jeho rozsah.

Obr. 5. Zapojení modulu převodníků

Vstupní napěťový rozsah je dán rezistorem R3 a odporem trimru P3, kterým se nastavuje zesílení operačního zesilovače IO8. Rezistor R5 má pouze ochranný charakter.

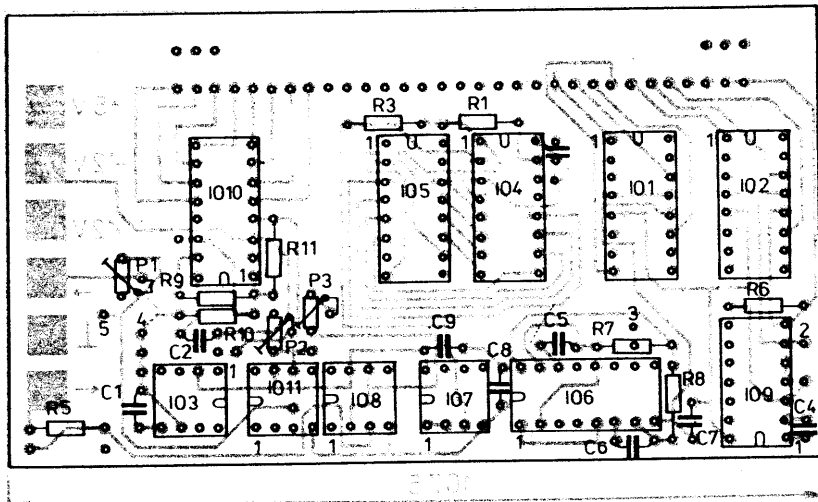
Výstupní signál komparátoru, přivedený na vstup D aproximačního registru, je následujícím hodinovým impulsem přepsán na sériový výstup D0. Takto je určen nejvýznamnější bit číselového signálu (MSB). Současně výstup Q7 přejde do úrovně L a celý cyklus se opakuje. S příchodem deváté náběžné hrany hodinového impulsu je určen nejméně významný bit (LSB). Přechod signálu CC z úrovně H do L ukončí dobu převodu a zapíše číselovou informaci z výstupu aproximačního registru do vyrovnávacích registrů, tvořených dvěma MH7475 (IO1 a IO2) kde je uchována po dobu dalších devíti hodinových impulsů. Doba převodu je 6,6 μ s. S příchodem náběžné hrany nového impulsu se aproximační registr opět spustí a celý proces se opakuje. Derivační člen, tvořený rezistorem R6 a kondenzátorem C4 spolu s MH7404 (IO9) vytváří impulsy do výstupního registru (obr. 7).

Převodník D/A

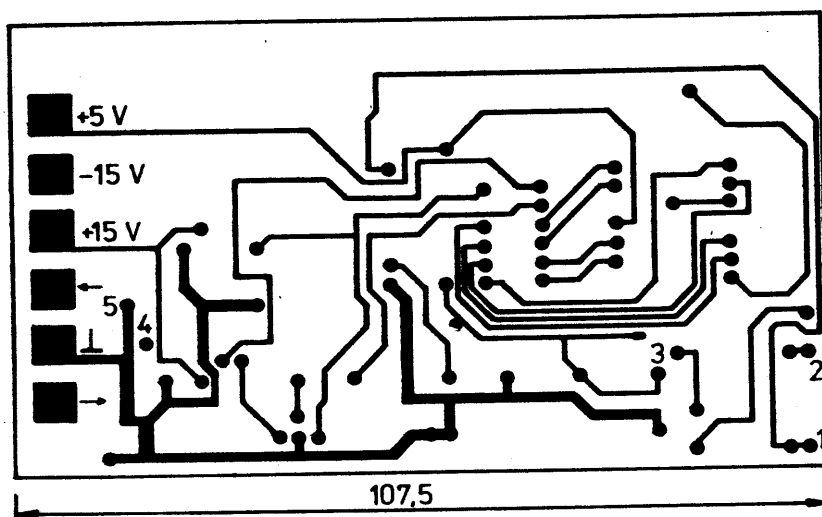
Převodník D/A je na této desce realizován monolitickým násobícím číselově-analogovým převodníkem s proudovým výstupem MDAC08 (IO10). Zapojení je v druhé části obr. 5. Výstupní proud převodníku D/A závisí jednak na osmibitovém vstupním slovu, jednak na vstupním referenčním proudu, nastaveném rezistory R9 a R10. Vzhledem k tomu, že tento obvod již byl popsán v (5), zmíním se o něm velmi stručně. Doba ustálení převodu je 150 ns. Proudový výstup z převodníku se pomocí rezistoru R11 a operačního zesilovače MA741CN (IO11) převádí na napěťový. Požadované výstupní napětí v bodě 5 nastavíme změnou zesílení operačního zesilovače. Odporový trimr P1 umožňuje definitivní nastavení výstupního napětí. Vzhledem k tomu, že stykový obvod MHB8255 je již vybaven vyrovnávací pamětí, nemusíme ji v tomto zapojení používat. Zdroj referenčního napětí, osazený IO3, je pro oba převodníky společný.

Konstrukční uspořádání

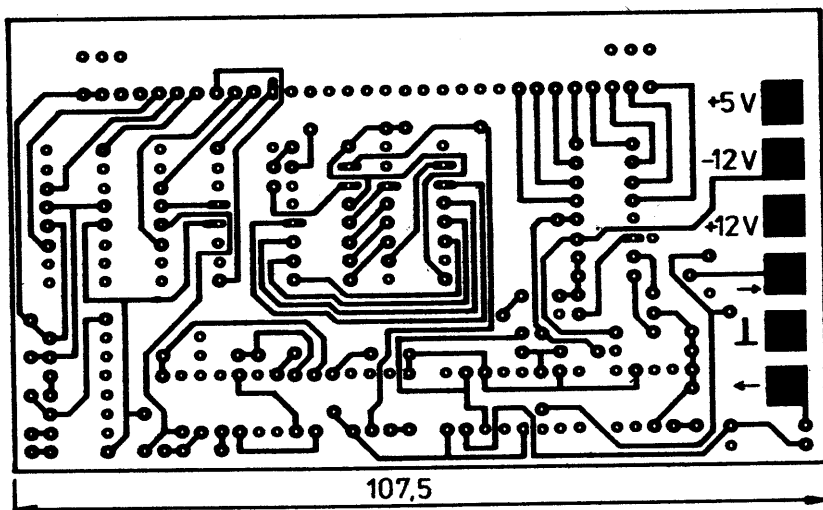
Oba převodníky jsou na společné desce o rozměrech 107,5 x 60 mm s oboustrannými plošnými spoji. V horní části desky je umístěn konektor FRB TY 5176211, kterým se připojí modul převodníků k modulu PIO. IO1 až IO3, IO7 a IO9 jsou připojeny k plošným spojům, IO4 až IO6, IO8, IO10 a IO11 jsou zasunuty do objímek. Vzhledem k tomu, že IO8 a IO11 mají po 2 x 4 vývodech a jsou umístěny na desce vedle sebe, lze pro ně použít jednu společnou objímku. Do měřicích bodů 1 až 5 lze připojovat dutinky z konektoru FRB, nebo pájecí špičky. Výstupní vývody jsou vytvořeny z plošek 5 x 5 mm a lze do nich zanytovat pájecí očka, nebo k nim vývody pouze připojovat. Rozmístění součástek na desce s plošnými spoji je na obr. 6, průběhy signálů na obr. 7, obrazec plošných spojů na obr. 8 a 9.



Obr. 6. Rozmístění součástek na desce s plošnými spoji W502 převodníků

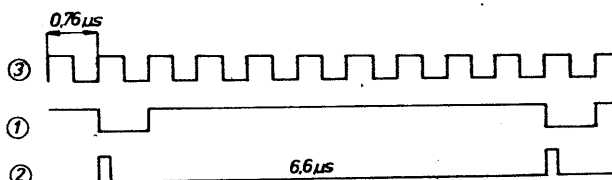


Obr. 8. Obrazec plošných spojů desky W402 převodníků — strana součástek



Obr. 9. Obrazec plošných spojů desky W502 převodníků — strana bez součástek

Obr. 7. Průběhy signálů v měřicích bodech



17

Jsou „posazeny“ do adresního prostoru původní paměti ROM a lze do nich přesunout operační systém. Přepínačem lze volit buď ROM s původním OS, nebo RAM s upraveným OS. Při umístění OS v RAM máme možnost (příkazy POKE) přepisovat obsahy paměťových buněk a tím si například předdefinovat tabulku znaků pro jiný druh písma, odstranit chyby v OS (např. NMI), nebo do volných a nevyužitých prostor uložit další programy, např. pro obsluhu tiskárny, dálkopisu atd. Vzhledem k malému příkonu paměti (při napájecím napětí 2V jednotky až desítky mikroampér), je na tomto modulu umístěn záložní zdroj, složený ze dvou kusů knoflíkových článků 1,5 V.

Po odpojení mikropočítače od sítě jsou tak paměti v pohotovostním režimu a program nebo data zůstávají uloženy v paměti i nadále. Po zapnutí počítače jsou opět k dispozici.

Popis zapojení

V popisovaném zapojení jsou jako IO2 a IO3 použity statické paměti typu RAM 6264 s organizací 8192 x 8 bitů. Vzhledem k tomu, že pro původní paměť ROM je vymezen prostor 16 kB, je nutné tyto paměti přepínat po blocích 8 kB. Paměť IO2 je adresována v prostoru 0 až 8 kB, paměť IO3 v prostoru 8 až 16 kB. Adresový dekodér je tvořen pamětí MH74S287 (IO1). Přepíná paměť RAM po 8 kB, dynamicky blokuje signál ROMCS a RAMCS. Umožňuje tři základní režimy, které jsou dány počtem paměti 6264, které máme k dispozici. Modul můžeme osadit pouze jedním kusem, nebo dvěma kusy. Pro všechny tři druhy provozu je již dekodér naprogramován a volí se pouhým přepojením vstupů E a F dekodéru na log. 0, nebo log. 1 (tab. 3).

Před zapnutím počítače je nutné předem rozepnout spínač S1. Dekodér přepne blok paměti do prostoru 48 až 64 kB (popř. 48 až 56, nebo 56 až 64 kB) paralelně k vnitřnímu dynamickým pamětím. Pomocným programem

Tab. 3. Programovací tabulka MH74287

vstup-adresa		výstup				rozsah
dec	hex	bin	Y4	Y3	Y2	
0	00	0000 0000	1	0	0	0 až 8 kB
1	01	0000 0001	0	1	0	8 až 16 kB
2	02	0000 0010	1	1	0	C
13	0D	0000 1101	1	1	0	C
14	0E	0000 1110	1	0	1	A
15	0F	0000 1111	0	1	1	0
16	10	0001 0000	1	1	0	0
17	11	0001 0001	0	1	0	1
18	12	0001 0010	1	1	0	0
30	1E	0001 1110	1	1	0	0
31	1F	0001 1111	0	1	1	0
32	20	0010 0000	1	0	0	1
33	21	0010 0001	1	1	0	0
45	2D	0010 1101	1	1	0	0
46	2E	0010 1110	1	0	1	0
47	2F	0010 1111	1	1	0	0
48	30	0011 0000	0	0	0	0
255	FF	1111 1111	0	0	0	0

Tab. 4. Pomocný program pro překlad OS do RAM

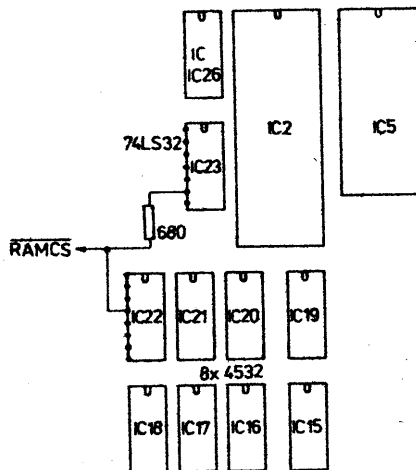
a) ve strojovém kódu mikroprocesoru Z80:
 8000 21 FF 3F LD HL, 3F FF
 8003 11 FF FF LD DE, FF FF
 8006 01 FF FF LD BC, FF FF
 8009 ED B8 LDDR
 800B C9 RET

b) v jazyce BASIC
 1 CLEAR 39000
 2 FOR n = 0 TO 11
 3 READ y: PRINT y
 4 POKE (4000 + n), y
 5 NEXT n
 6 DATA 33, 255, 63, 17, 255, 255, 1, 255, 255, 237, 104, 201

(Tab. 4) se blok prvních 16 kB paměti ROM přemístí do posledních 16 kB paměti RAM (48 až 64 kB). Sepnutím spínače S1 (vstup D = log. 0) dojde k přepnutí dekodéru do rozsahu 0 až 16 kB, k dynamickému zablokování paměti ROM signálem ROMCS = log. 1. a k odblokování vnější paměti RAM v rozsahu 0 až 8 a 8 až 16 kB.

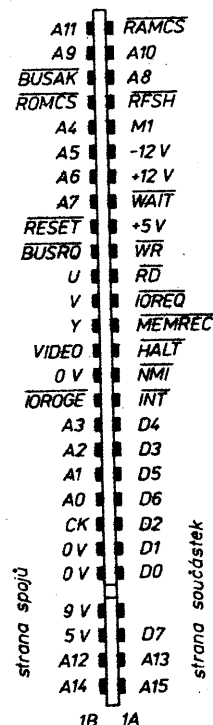
Po tomto přepnutí původní program zůstane i nadále v posledních 16 kB (48 až 64 kB). Lze jej smazat nebo nahradit jiným programem. Rozpojíme-li opět spínač S1 a přepneme vnější paměť zpět, je nutné (pokud jsme před tím provedli RESET, vypnutí počítače, nebo došlo ke zhroucení programu s následnou inicializací počítače) předem zadat CLEAR 47000 (posunutí RAMTOP a vymezení prostoru pro data).

V předchozím popisu funkce nebylo nutné zasahovat do počítače. Budeme-li ale chtít po připojení vnější paměti RAM do prostoru 48 až 64 kB blokovat vnitřní paměť RAM a v ní zachovat původní obsah (data nebo programy ve strojovém kódu), musíme zavést dynamické blokování paměti RAM signálem RAMCS. Mikropočítač ZX-Spectrum na rozdíl od svého předchůdce ZX-81 nemá tento signál vyveden. Bude to tedy znamenat zásah do počítače. Tím získáme jednu provázku možnost zablokovat vnitřní paměť 32 kB (tvořenou osmi obvody 4532) a připojovat si na sběrnici pomocí konektoru další bloky paměti RAM nebo EPROM a ty pak libovolně podle potřeby přepínat. Přeřídíme spoj vedoucí z vývodu 3 a 4 IO23 (SN74LS32) na vývod 4 bloku paměti IO15 až IO22 (4532) a nahradíme ho rezistorem 680 Ω.

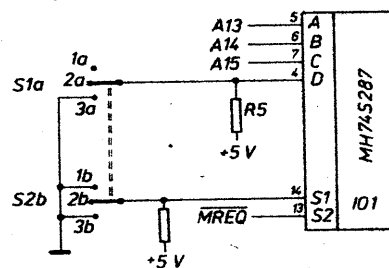


Obr. 14. Úprava ZX-Spectrum pro vyvedení RAM CS

Vývod rezistoru směrem k pamětím 4532 propojíme s volným kontaktem výstupního konektoru. Při pohledu ze zadu na ZX-Spectrum je to v horní řadě na levé krajní pozici (obr. 14 a 15).



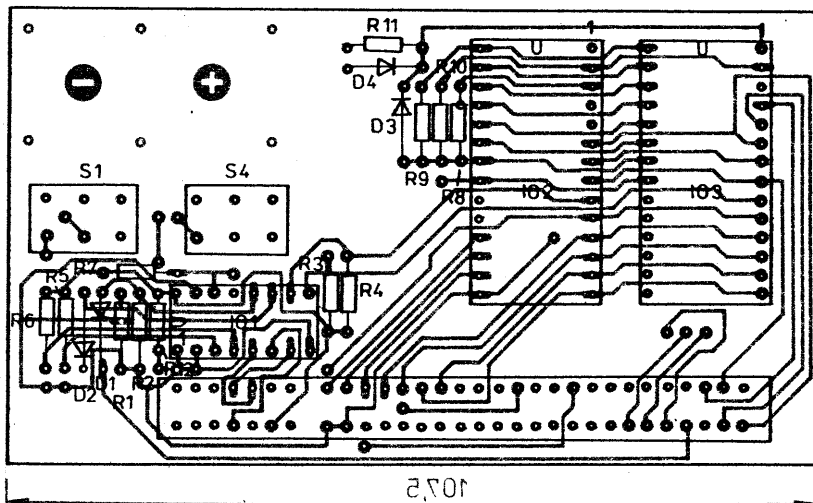
Obr. 15. Vyvedení RAM CS na konektor mikropočítače



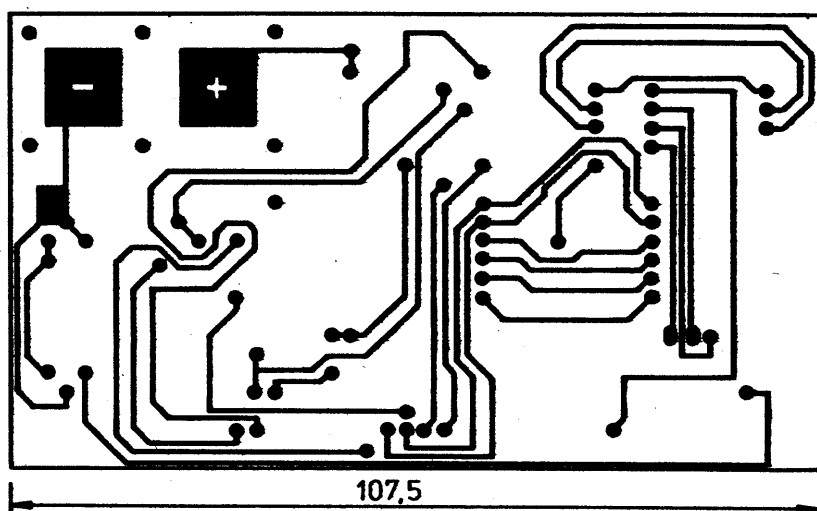
Obr. 16. Náhrada dvou spínačů jedním dvoupólovým přepínačem

Správnou funkci dekodéru a blokování si ověříme. Do objímky IO1 zasuneme naprogramovanou paměť MH74S287, objímky IO2 a IO3 zůstanou volné. Paměť MH74S287 je přepojena do režimu 48 až 64 kB. Po zapnutí počítače zadáme příkaz PRINT PEEK 23732 + 256 * PEEK 23733. Na obrazovce se objeví 49151, což je konec paměti RAM. Objeví-li se 65535, je někde chyba a máme ji odstranit. Po zasunutí obou pamětí RAM se po zadání předchozího příkazu má správně objevit 65535. To znamená, že počítač s novou pamětí komunikuje.

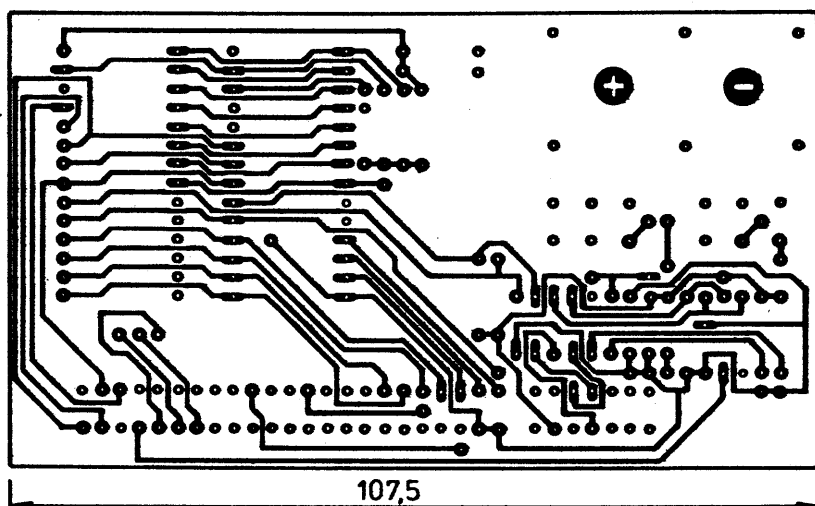
Po této úpravě při zapojení vnější paměti do rozsahu 48 až 64 kB bude vždy vnitřní paměť v tomto rozsahu zablokována a její obsah zachován. Budeme-li chtít zablokovat vnější paměť RAM, rozepneme spínač S3. Log. 1 na uvolňovacím vstupu S1 MH74S287 zablokuje dekodér a výstupy přivede do třetího stavu. Spínač S1 a S3 lze nahradit jedním třípolohovým dvoupólovým přepínačem (Obr. 16), který umožní v poloze 1 rozsah 48 až 64 kB, v poloze 2 dojde k zablokování vnější paměti a v poloze 3 rozsah 0 až 16 kB.



Obr. 11. Rozmístění součástek na desce s plošnými spoji W503 modulu RAM



Obr. 12. Obrazec plošných spojů desky W503 modulu RAM — strana součástek



Obr. 13. Obrazec plošných spojů desky W503 modulu RAM — strana bez součástek

Jak již bylo řečeno v úvodní části popisu, lze během provozu počítače měnit pomocí příkazu POKE obsah vnější paměti RAM i po přepnutí do rozsahu 0 až 16 kB. Zde však u prvního bloku 0 až 8 kB dochází k vynulování

paměťových buněk na adresách A0 až A4, které je zřejmě způsobeno neúplným dekódováním adresové sběrnice v ULA, která generuje v tomto rozsahu signál řídící sběrnice RD/WR. Protože se předpokládá, že je zde připojena

paměť, ze které se pouze čte, nevadí to. Nahradíme-li však paměť ROM pamětí RAM, začne to vadit. Nedostatek lze odstranit úplným dekódováním adresové sběrnice (všech šestnáct vodičů) a zablokováním signálu WR v tomto rozsahu. To však znamená použít řadu IO navíc.

Tento problém lze jednoduše obejít tak, že pro provoz vnější paměti RAM v rozsahu 0 až 8 kB zvolíme spínačem S4 (rozepnut) WR = log. 1 režim čtení. Režim zápisu (S4 je sepnut) zvolíme přepnutím S1 do rozsahu 48 až 56 kB. Příkazy POKE budeme používat pouze v tomto rozsahu. Má to i tu výhodu, že při rozepnutém spínači S4 chráníme obsah paměti před vymazáním. Signál WR lze přepínat pouze pro první paměť RAM v rozsahu 0 až 8 kB a druhou mít trvale připojenou na signál WR a zapisovat do ní i v rozsahu 8 až 16 kB, nebo signál WR spínačem S4 přepínat pro obě společně a i do druhé vstupovat pouze v rozsahu 56 až 64 kB. V tab. 2 jsou uvedeny požadované funkce spínačů.

Tab. 2.

S1	stránka rozsahu 0 až 16, 48 až 64 kB
S2	propojka — nastavení pozice paměti RAM, jedna, nebo dva kusy
S3	zablokování dekóderu a paměti RAM
S4	zablokování zápisu do vnějších pamětí RAM

Na pozici IO1 je ve funkci dekóderu paměť PROM MH74S287 s třístavovým výstupem. To znamená, že při neaktivním signálu ROMCS a RAMCS rezistory R1 a R2 zajišťují na výstupech Y1 a Y2 log. 0 a při neaktivním signálu CS1 a CS2 rezistory R3 a R4 na výstupech Y3 a Y4 log. 1. Rezistory R5 až R8 při rozpojených spínačích zajišťují na příslušných vstupech log. 1. Rezistor R11 umožňuje dobíjení napájecího článku v době provozu počítače. Diody D1 a D2 oddělují signál ROMCS a RAMCS, vystupující z počítače. Dioda D3 zabráňuje vybíjení článku při vypnutí počítače a D4 zabráňuje propojení napájecího napětí na napájecí článek.

Programovací tabulka je v Tab. 3. Pro režim 0 až 16 kB je program na adresách 0 až 15, a na vstupy E až H obvodu MH74S287 je přivedena log. 0 (spojeno se zemí).

Pro režim 8 až 16 kB je program na adresách 16 až 31, na vstup E je přivedena log. 1 přes rezistor R6 a na vstupy F až H log. 0.

Pro režim 0 až 8 kB je program na adresách 32 až 47, na vstup F je přivedena log. 1 přes rezistor R6 (volí se propojkou popř. přerušením plošného spoje) a na vstup E a G až H log. 0. Zbývající prostor na adresách 48 až 255 zůstává volný a lze ho případně použít pro jinou dekódovací tabulku.

Konstrukční provedení

IO1 až IO3 jsou zasunuty v objímkách. Ostatní součástky jsou zapájeny v plošných spojih. Knoflíkové články typu S105 ø 16×6 mm jsou zasunuty ve dvou pouzdrech ø 18/1,5 mm a výška 5,5 mm, která jsou podle obr. 11 přilepena lepidlem EPOXY 1200 k plošnému spoji. Bronzový pásek

o tloušťce 0,4 mm, který je k základní desce připevněn dvěma šrouby M2, drží tyto články v pouzdech a zajišťuje jejich vzájemné propojení.

Přepínače S1, S3 a S4 jsou miniaturní, a lze je zakoupit v NDR. K plošnému spoji jsou připevněny úhelníkem z hliníkového plechu. S3 je tvořen propojením, popř. přerušením plošného spoje.

Celý modul RAM je k mikropočítači ZX-Spectrum připojen konektorem WK 46 580, nebo přes redukci konektorem FRB TY 517 6211.

Schéma zapojení je na obr. 10, rozmístění součástek na obr. 11, obrazec plošných spojů na obr. 12 a 13, úprava počítače ZX-Spectrum na obr. 14 a zapojení výstupního konektoru na obr. 15. Na obr. 16 je schéma zapojení jednoho přepínače místo S1 a S3.

Programové vybavení

K tomuto modulu byly odladěny dva programy, které umožňují spolupráci modulu RAM s mikropočítačem (program v Tab. 5) a demonstrují možnost použití (Tab. 6).

První část programu, uvedeného v Tab. 5, po zadání příkazu RUN přeloží OS do paměti RAM v rozsahu 48 až 64 kB. Tento překlad se provede i v případě, kdy není k mikropočítači připojen vnější modul RAM. Je-li připojen, je nutné, aby byl spínač S4 sepnut a S1 přepnut do rozsahu 48 až 64 kB. Po překladu se vytiskne v prvním sloupci desítkový výpis obsahu bloku prvních 16 kB, včetně adres, v druhém výpis přeloženého bloku včetně adres + 49152. Tuto část programu lze kdykoliv použít (po zadání RUN 15) ke kontrole obsahu paměti RAM v případech, kdy došlo ke zhroucení programu a nemáme jistotu, že původní obsah paměti zůstal zachován. Přepínejeme-li příkazem POKE některé adresy, můžeme je tímto způsobem kontrolovat.

Třetí část programu (zadání RUN 40) porovnává obsah jednotlivých adres v rozsahu 0 až 16 a 48 až 64 kB a tiskne odchylky. Oceníme ji v případech, kdy nechce mikropočítač s modulem spolupracovat. Tento případ může nastat při neopatrné manipulaci s S4 v rozsahu 0 až 16 kB, nebo při provozu ze záložní baterie (modul k ní byl delší dobu připojen a došlo k poklesu napětí), kdy může dojít k chybám na některých adresách a potřebujeme rychlou kontrolu. Při provozu se též ukázalo potřebné v některých případech ponechat počítač s připojeným modulem po zapnutí několik desítek sekund v klidu a pak s ním teprve komunikovat.

Program v Tab. 6 na jednoduchém příkladu demonstruje použitelnost tohoto modulu. Umožňuje předefinovat tabulku méně často používaných symbolů a značek za písmena české abecedy. Zapisuje se stejně, jak bylo uvedeno v předchozí části. Volbou bloku přepínače S1 lze volit OS s původní, nebo s opravenou abecedou.

Seznam součástek

Polovodiče:	
IO1	MH74S287
IO2	6264
IO3	6264
VD1, VD2	
VD4	KA 261, 262, 501
VD3	KY 130

Rezistory:	
R1, R2	470 Ω, TR112
R3, R9	5,6 kΩ, TR112
R10	12 kΩ, TR112
R11	4,7 kΩ, TR112

Konstrukční části:
konektor WK 46580 1 kus
objímka 2x 8 vývodů GAF49769 1 kus
objímka 2x 14 vývodů TX 787 522 2 kusy
deska s plošnými spoji oboustranná 1 kus
knoflíkový článek 1,5 V typ S105 2 kusy
přepínač miniaturní páčkový 3 kusy

Napájecí zdroj

Přesto, že v minulých číslech AR se objevila celá řada návodů na napájecí zdroje, uvádím zde pro úplnost popis jednoduchého a univerzálního zdroje pro napájení modulu převodníků. Po úpravě (vyvedení napětí +9 V) by bylo možné využít ho k napájení celého mikropočítače.

Popis zapojení

Popisovaný zdroj je osazený třemi integrovanými stabilizátory typu MA 78xx, napájenými ze dvou samostatných usměrňovacích bloků. Tyto bloky jsou osazeny dvěma dvojicemi usměrňovacích diod typu KY940 a KY950. V případě většího odběru a nebezpečí přehřívání je možné opatřit je chladiči. Tyto diody jsou schopny s chladičem snést proud maximálně 3 A.

Jeden blok napájí stabilizovaný zdroj +5 a +15 V (popř. +9 V), druhý -15 V. Dioda D5 indikuje zapnutí zdroje a je připojena přes svorku č. 7 na první blok zdroje. Podle potřeby ji lze připojit přímo na výstup některého stabilizátoru (přes rezistor).

Konstrukční uspořádání

Napájecí zdroj je umístěn včetně síťového transformátoru na jedné desce s plošnými spoji. V uvedeném konstrukci byl použit síťový transformátor s jádrem EI 25 x 25 a jeho navijecí předpis je na obr. 18. K desce s plošnými spoji je přišroubován přes distanční sloupky šrouby M4. Vývody jednotlivých vinutí jsou připojeny ke svorkám 1 až 6. IO1 a IO2 jsou umístěny na jednom společném chladiči, IO3 je na druhém. Stabilizátory jsou společně s chladiči přišroubovány přes distanční sloupky k desce s plošnými spoji a jejich vývody jsou přímo připájeny do plošných spojů.

Tab. 5. Překlad OS do RAM

```

1 REM PREKLAD OS DO RAM
2 CLEAR 39000: PRINT " P R E K L A D O S "
3 FOR N=0 TO 11
4 READ Y: PRINT Y;" "
5 POKE (40000+N),Y
6 NEXT N
7 DATA 33,255,63,17,255,255,1,255,255,237,184,201
8 RANDOMIZE USR 40000
10 PAUSE 50: CLS
12 STOP
15 PRINT "V Y P I S P A M E T I"
18 PRINT
20 PRINT "-----"
22 PRINT
25 FOR N=0 TO 16383
30 PRINT N;" ";(N+49152);" ";PEEK N;"
";PEEK (N+49152)
35 NEXT N
36 STOP
40 PRINT " T E S T C H Y B Y "
41 PRINT "-----"
42 FOR N=0 TO 16383
43 LET C=N+49152
44 LET A=PEEK N
45 LET B=PEEK C
46 IF A=B THEN GO TO 48
47 PRINT N;" ";C;" ";PEEK N;" ";PEEK C
48 NEXT N
49 PRINT " TISK S100"

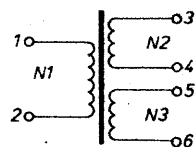
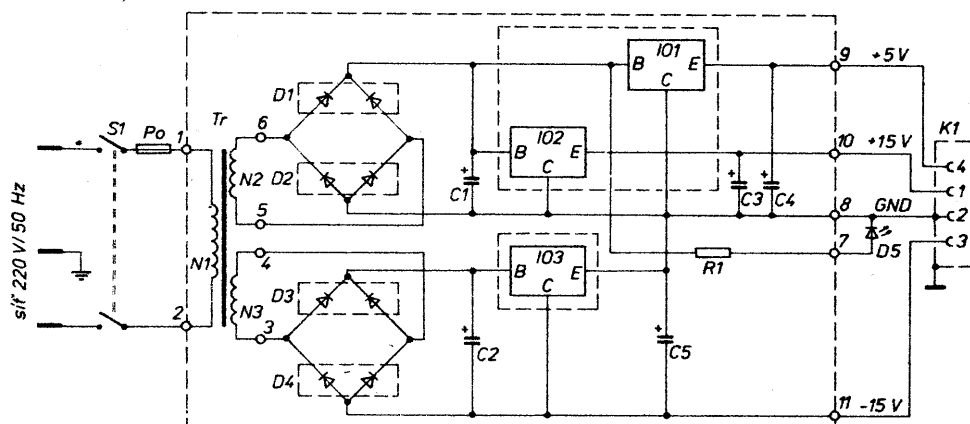
```

Tab. 6. Nová tabulka znaků

```

1 PRINT " NOVA TABULKA ZNAKU"
2 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
3 FOR N=64776 TO 64847
4 READ A
5 POKE N,A
6 NEXT N
7 DATA 8,16,56,4,60,68,60,0,0,36,36,0,0,0,0,0,40,16,56,68,120
,64,60,0,0,8,62,40,62,10,62,8,8,16,0,48,16,16,56,0,8,16,68,68,68,
60,4,56,8,16,56,68,68,68,56,0,8,16,56,68,120,64,60,0,40,16,28,32,
32,32,32,0
10 FOR N=65024 TO 65031
12 READ B
14 POKE N,B
16 NEXT N
18 DATA 8,16,56,68,120,64,60,0
20 FOR N=65240 TO 65279
22 READ C
24 POKE N,C
26 NEXT N
28 DATA 40,16,28,32,32,32,0,80,36,4,60,68,68,60,0,24,24,68,
68,68,56,0,0,16,56,84,16,16,16,0,8,16,68,68,68,68,56,0
30 FOR N=65496 TO 65535
32 READ D
34 POKE N,D
36 NEXT N
38 DATA 40,16,16,56,16,16,12,0,40,16,56,64,56,4,120,0,40,16,12
4,8,16,32,124,0,40,16,120,68,68,68,68,0,40,16,28,32,32,32,28,0
40 FOR N=54592 TO 54599
41 READ X
42 POKE N,X
43 NEXT N
44 DATA 32,68,111,108,101,125,97,108
46 PRINT
48 PRINT " PRELOZENO "
49 PRINT " TISK S100"

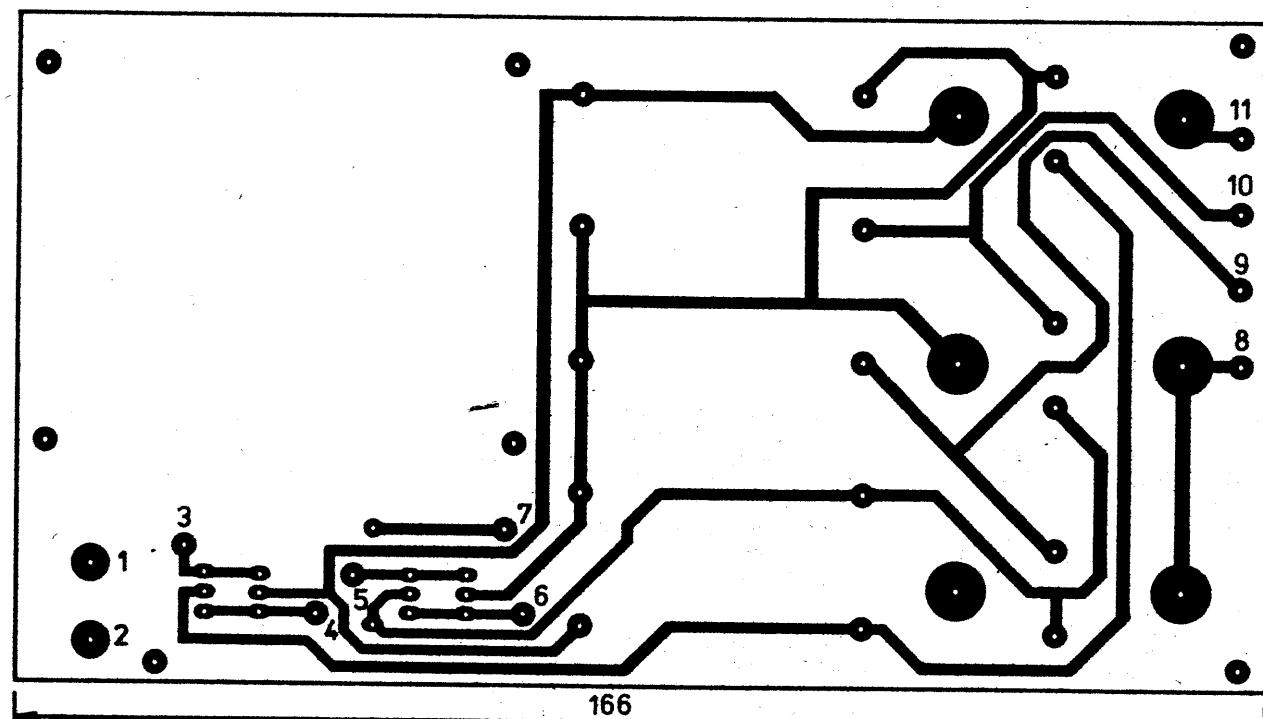
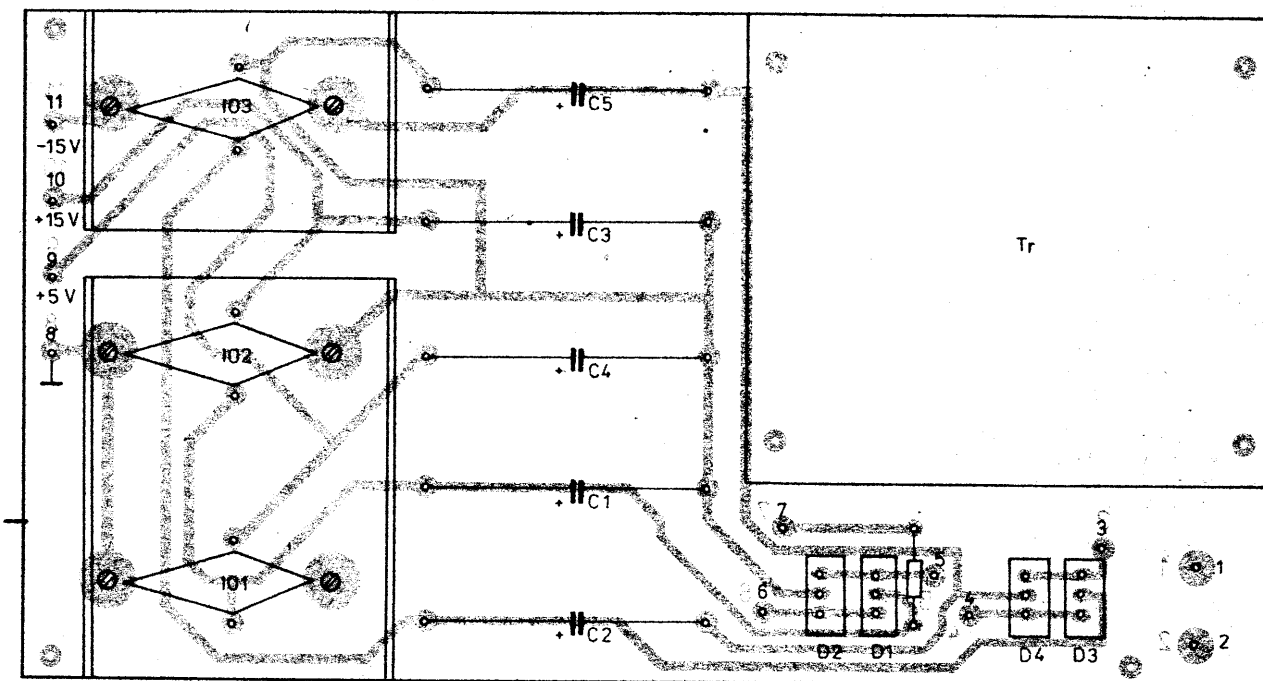
```



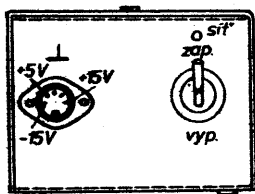
Obr. 18. Navíjecí předpis síťového transformátoru

jádro: EI 25 x 25 mm
 N1: 220 V — 1580 z vod. \varnothing 0,18 CuPL
 N2: 18 V/1,8 A — 160 z, \varnothing 0,95 CuPL
 N3: 18 V/1 A — 160 z, \varnothing 0,65 CuPL

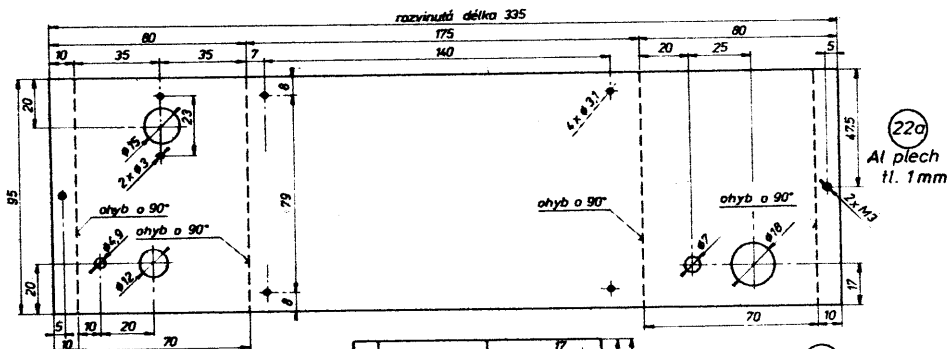
Obr. 17. Schéma zapojení zdroje



Obr. 19, 20. Rozmístění součástek a obrazec plošných spojů na desce zdroje W504

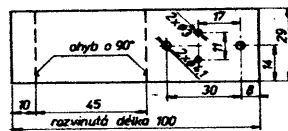


Obr. 21. Čelní panel zdroje

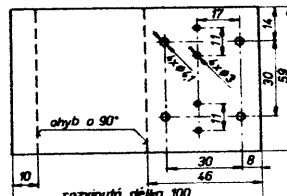


Obr. 22. Mechanické díly zdroje

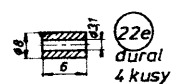
a — spodek skříňe, b — vrch skříňe,
c — chladič I, d — chladič II,
e — distanční sloupky, f — distanční
sloupky transformátoru, g — distanční
podložka IO



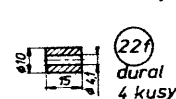
22c Al plech tl. 1mm



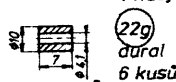
22d Al plech tl. 1mm



22e dural 4 kusy



22f dural 4 kusy



22g dural 6 kusů

Deska s plošnými spoji je umístěna ve skříňce z hliníkového plechu a přes distanční sloupky k ní přišroubována. Na předním panelu je na levé straně konektor K1 (pětikolíková magnetofonová zásuvka) pro výstupní napětí, na pravé síťový vypínač spolu se signální diodou. Pojistka Pj je na zadní stěně. Schéma zapojení zdroje je na obr. 17, obrazec plošných spojů a rozmístění součástek na obr. 19 a 20, celková sestava na obr. 21 a výkresy mechanických dílů na obr. 22.

Mechanické díly

Skříňka, složená ze dna a krytu, je vyrobena z hliníkového plechu a tloušťce 1mm. Po orýsování vyvrtáme všechny otvory a díly ohneme do potřebného tvaru. U dna se nejdříve ohnou okraje a pak teprve přední a zadní stěna. Kryt je přišroubován ke dnu dvěma šrouby M3 a to shora. Skříňku lze po důkladném mechanickém očištění mořit v hydroxidu sodném (pozor na poleptání pokožky), nebo příslušné díly nastříkat vhodnou nitrobarvou (SPRAY). Chladiče jsou též z hliníkového plechu.

Seznam součástek

Polovodičové součástky:

IO1	MA7805
IO2, IO3	MA7815
D1, D3	KY940
D2, D4	KY950
D5	LQ1731

Kondenzátory:

C1, C2	500 μ F/35 V, TE986
C3 až C5	500 μ F/15 V, TE984

Rezistory:

R1	2,2 k Ω , TR 212, 151 atd.
----	-----------------------------------

Konstrukční díly:

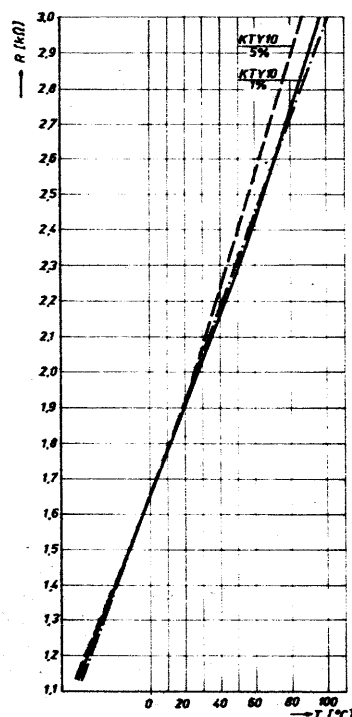
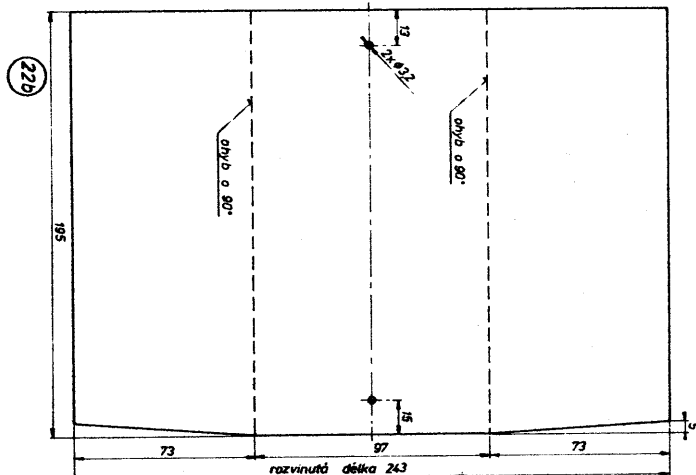
síťový vypínač 250 V/4 A — dvoupólový
pojistkový spodek REMOS 6 A
trubičková pojistka 0,3 A
chladič zdroje +5 V, +15 V

chladič zdroje -15 V
deska s plošnými spoji
skříňka kompletní
distanční sloupek pod síťový transformátor 4 ks
distanční sloupek pod desku 4 ks
distanční sloupek pod stabilizátor 6 ks
spojovací materiál (šrouby atd.)
síťový transformátor
flexošňůra třípramenná 6 A
konektorová zásuvka

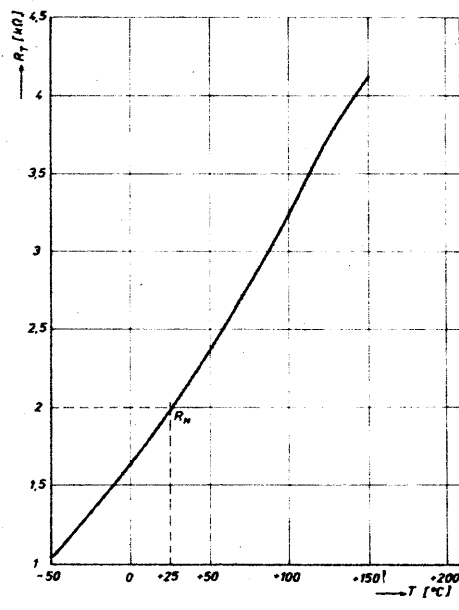
5. Měření teploty

Jako doplněk k modulu převodníků byl navržen a sestaven snímač teploty, kde ve funkci čidla byl použit moderní senzor.fy Siemens KTY10. Výhoda tohoto senzoru spočívá v téměř lineárním průběhu v rozmezí -50 až +150 °C a v poměrně velké změně odporu v tomto rozsahu.

Prodává se ve dvou třídách přesnosti, s odchylkou 5 % a 1 %. Na obr. 28 je uvedena kalibrační křivka verze 5 % jak ji dodává fy Siemens, na obr. 27 je pro porovnání uvedena křivka senzoru (zřejmě 1 %), který jsem měl k dispozici.



Obr. 27. Kalibrační tabulka senzoru KTY 10 — 5 %, 1 %



Obr. 28. Kalibrační křivka použitého senzoru

Seznam součástek

Polovodičové součástky
 OZ MAA741, MAA741 CN
 Rx KTY10

Rezistory:
 R1 68 kΩ, TR162
 R2 12 kΩ, TR162
 R3 1,8 kΩ, TR162
 R4 10 kΩ, TR162
 R5 220 Ω, TP095
 R6, R7 100 kΩ, TR161, TR212
 R8 100 kΩ, TP095
 R9, R10 3,3 kΩ, TP095

Kondenzátory
 C1 0,1 μF, TK783

Konstrukční části:
 deska s plošnými spoji

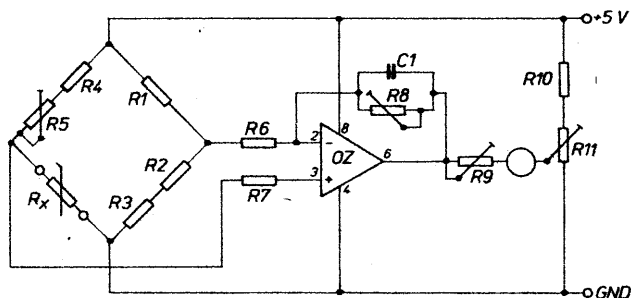
Příklad použití

Z popsaného modulu lze ve spojení s modulem převodníků realizovat jednoduchý mikropočítačový teploměr.

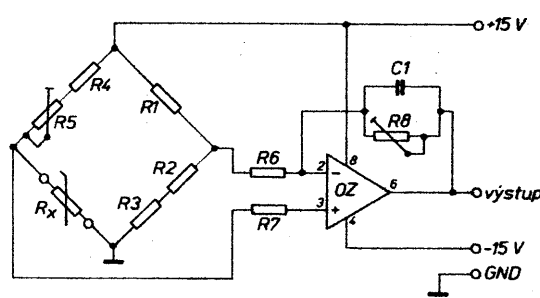
Bude-li obvod nastaven pro rozsah 0 až 100 °C, což bude odpovídat výstupnímu napětí 0 až 10 V, bude pak 1 bit = 0,04 V = 0,4 °C. Konstanta pro převod bude $K = 0,4$. Programově z portu přečtenou hodnotu vynásobíme touto konstantou a dostaneme teplotu již v °C.

Pro uvedený obvod byl odladěn program v jazyce BASIC (Tab. 7), který se skládá ze dvou částí.

V první části programu (po startu se stiskne 0) se čtou hodnoty z portu převodníku A/D a ukládají se příkazem POKE do paměti. Vzhledem k tomu, že údaj portu je osmibitový, stačí k jeho uložení do paměti pouze jedna adresa — z toho plyne, že je nutné vždy určitou část paměti vyhradit pro data, která se ukládají do adresy 65533 směrem



Obr. 23. Schéma analogového teploměru



Obr. 24. Schéma zapojení modulu k měření teploty

Z uvedených průběhů vyplývá, že v menším rozsahu teplot (0 až 100 °C) —50 až +50 °C lze dosáhnout velmi dobré linearity. Při měření uvedeného vzorku byla pro teploty 0 °C použita ledová tříšť, pro vyšší teploty různé teplá voda. K měření teploty jsem použil termistorový teploměr PU 391/1, k měření odporu digitální multimetr M1T242. Maximální proud senzorem je 20 mA.

Existenci tohoto levného senzoru (1 až 2 DM) začala nová éra měření teplot pomocí přenosných multimetrů a digitálních voltmetrů. Stačí totiž takový multimetr rozšířit a zdroj konstantního proudu (stačí na to dva tranzistory) a přímo, bez jakékoliv linearizace měřit úbytek napětí na senzoru, který udává teplotu s přijatelnou přesností. Na obr. 23 uvádím jednoduché zapojení přímo ukazujícího teploměru s jedním operačním zesilovačem.

Popis zapojení

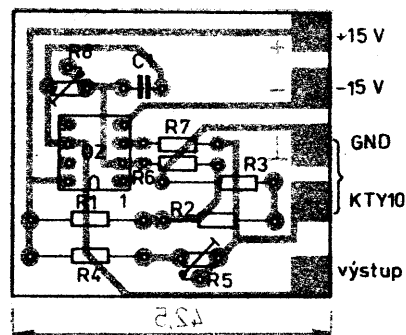
Senzor KTY10 je zapojen v jedné větvi Wheatstonova můstku, který je napájen kladným napětím +15 V. Můstek se zhruba vyváží rezistory R2 a R3, jemně pak nastavitelným rezistorem R5. Výstupní napětí se přivádí přes rezistory R6 a R7 na vstup 2 a 3 operačního zesilovače. Rezistor R8 slouží k nastavení zesílení zesilovače. Kondenzátor C1 zvětšuje stabilitu obvodu. Schéma zapojení je na obr. 24, rozmístění součástek na desce s plošnými spoji na obr. 25 a obrazec plošných spojů na obr. 26.

Nastavení můstku

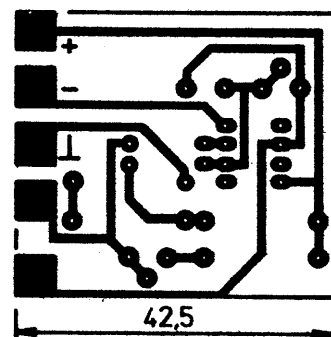
Změříme odpor senzoru při teplotě 0 °C a pak pokud možno při největší teplotě (80 až 100 °C). K nastavení obvodu použijeme dva nastavitelné rezistory 3,3 kΩ, typu TP 095 apod. Rezistory si označíme 0 °C a např. 80 °C. Nastavíme je nejlépe na digitálním ohmmetru na odpor, odpovídající daným teplotám. Na vstupní svorky plošného spoje, určené pro připojení senzoru KTY 10, připojíme střídavě oba rezistory. Na vývod 6 operačního zesilovače připojíme voltmetr a připojíme modul na napájecí napětí ±15 V. Změnou odporu rezistorů R2, R3 a R5 nastavíme nulovou výchylku (pro 0 °C). Zaměníme rezistor za druhý odpovídající např. 80 °C, a rezistorem R8 nastavíme výstupní napětí na +8 V. Tento postup je nutné několikrát opakovat, až jsou odchylky minimální. Definitivně nastavíme obvod po připojení k desce převodníků trimrem P3.

Konstrukční uspořádání

Obvod pro měření teploty je umístěn na desce s plošnými spoji a je propojen s modulem převodníků čtyřpramenovým plochým vodičem. Pro připojení senzoru KTY10 jsou použity dutinky z konektoru FRB. Dvoupramenový kabelík, na jehož konci je připájen senzor, má na druhém konci dvě špičky z konektoru FRB tak, že tvoří miniaturní zástrčku.



Obr. 25. Rozmístění součástek na desce W505 teploměru



Obr. 26. Obrazec plošných spojů desky W505 teploměru

nahoru. V programu to zajišťuje příkaz CLEAR 45000. Proto pokyn programu pro zadání rozsahu paměti slouží spíš pro definování počtu měření tak (nesmí přesáhnout povolený rozsah paměti), aby nedošlo ke zhroucení programu.

Hodnoty z portu se čtou každých 60 s (nastaveno příkazem PAUSE 3000). Změnou tohoto příkazu lze tuto dobu změnit. Odchylka po 24 hodinách činila 30 s. Tento čas se zároveň zobrazuje v levé části obrazovky. Proto je nutné po startu programu vždy zadat současný čas v hod. a minutách, který se uloží na adresy 65534 a 65535, kde je k dispozici pro zpětné čtení obsahu paměti.

Druhá část programu (po zastavení se opět zadá RUN a stiskne 1) přečte počáteční čas (zobrazený v levé části obrazovky), a hodnoty od adresy 65533 směrem nahoru, které po vynásobení konstantou se zobrazí v pravé části obrazovky již jako skutečné teploty v °C.

Tento program názorně demonstruje použitelnost uvedené sestavy modulů s mikropočítačem při měření analogových hodnot. Vzhledem k tomu, že zůstává kanál C volný, lze ho využít např. pro přepínání různých měřicích bodů a uvedené sestavy použít pro snímání teplot, nebo jiných veličin při různých měřeních v rozsáhlejších systémech. Osvědčilo se použití modul RAM 16 kB (popsaný v části 3), do kterého (přepnut do rozsahu 48 až 64 kB a napájen ze záložní baterie) se ukládala data z měření. Při výpadku napájení počítače tato data zůstala v paměti zachována. Je ovšem nutné před opětovným zapnutím mikropo-

Tab. 3. Obsluhový program pro měření teploty

```

1 REM HODINY, MERENI TEPLoty a ULOZENI DO PAMETI
2 OUT 127,144
3 CLEAR 45000:LET B=0
4 PRINT "ZVOL CTEMI - 1, NEBO ZAPIS - 0":INPUT F
5 IF F=1 THEN GO TO 40
6 PRINT "ZADEJ ROZSAH PAMETI":INPUT C:PRINT C,65533-C
7 PRINT "ZADEJ CAS: HODINY A MINUTY"
8 INPUT D:INPUT E
9 POKE 65535,D:POKE 65534,E
10 PRINT "          CAS          TEPLOTA"
11 FOR H=D TO 23
12 FOR M=E TO 59
13 LET B=B+1
14 LET X=IN 31
15 POKE (65534-B),X
16 PRINT B;" - ";H;" ":";M;"0.4*PEEK (65534-B)
17 PAUSE 3000
18 IF C=B THEN STOP
19 POKE 23692,100
20 NEXT M
21 LET E=0
22 NEXT H
23 LET D=0
24 GO TO 10

35 REM CTEMI PAMETI
40 PRINT "          CAS          TEPLOTA"
41 LET D=PEEK 65535:LET E=PEEK 65534
42 FOR H=D TO 23
43 FOR M=E TO 59
44 LET B=B+1
45 PRINT B;" - ";H;" ":";M;"0.4*PEEK(65534-B)
46 NEXT M
47 LET E=0
48 NEXT H
49 LET D=0
50 GO TO 42

```

čítače rozpojit S4 (blokování signálu WR).

Seznam použité literatury

- [1] AR-A 6/1985.
- [2] AR-A 2/1987.
- [3] ST 5/1986.
- [4] Katalog el. součástek 1983 + 84.
- [5] AR-A 9 až 12/1983.
- [6] Elektor 9/1984.
- [7] Servisní dokumentace pro ZX-Spectrum.
- [8] Logan, J., O'Hara, F.: The Complete Spectrum ROM Disassembly.
- [9] Siemens Componens 18 — 1980/5, str. 241 až 245.
- [10] Siemens Componens 20 — 1982/4, str. 117 až 122.

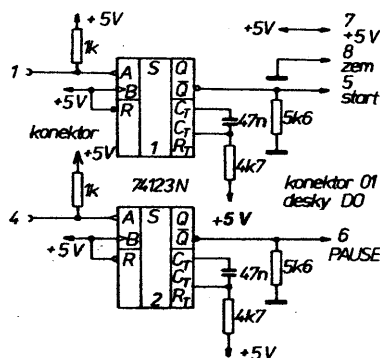
Připojení magnetofonu SM 260 jako paměťové jednotky k SAPI

RNDr. Ing. Jaroslav Lhota

Vzhledem k tomu, již delší dobu TESLA ELTOS DIZ nedodává k souboru SAPI kazetopáskovou jednotku jako vnější paměť, jsou uživatelé nuceni si ji opatřovat sami. Protože zápis dat na pásku se děje s dosti velkou rychlostí 2400 Bd, je lépe zvolit kvalitnější kazetový magnetofon, jako např. TESLA SM260. Pro připojení magnetofonu lze použít poněkud upravený kabel KB-05 dodávaný jako příslušenství SAPI a dále je nutno doplnit ovládací obvody magnetofonu, aby bylo možno softwarově ovládat funkce „START“ a „PAUSE“.

Pro vstup dat z desky DSM-1 mikropočítače SAPI do magnetofonu je vhodné použít vstup LINE (u starších výrobků ozn. UNIV), který povoleným rozsahem plně vyhoví.

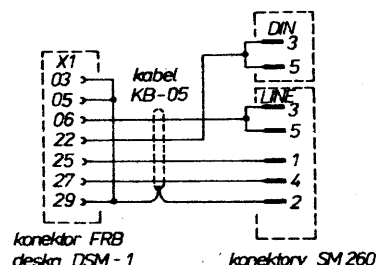
Přenosu dat z magnetofonu do počítače rovněž bez úprav vyhoví konektor DIN (dříve RADIO).



Obr. 1.

Pro dálkové ovládání lze s výhodou použít kompletního elektronického ovládání všech funkcí, jež je v magnetofonu použito. Ovládací signál z desky DSM-1 ve formě úrovně (původně sloužící ke spínání napájení magnetofonu) je nutné převést na signál ve tvaru záporného impulsu, neboť přivedením úrovně na desku přepínačů by byla blokována činnost ostatních tlačítek pro ruční ovládání. To je realizováno jedním integrovaným obvodem 123 N (v poslední době byla dostupná sovětská verze K 155AG 3) v zapojení pole obr. 1. Časová konstanta obvodu je nastavena tak, aby po přivedení nulové úrovně na vstup A byl vygenerován záporný impuls délky asi 0,2 ms,

což plně vyhoví ke spuštění příslušné funkce. Signál je přiveden paralelně k ovládacím tlačítkům a neomezuje jejich používání. Pro přenos ovládacích signálů ze SAPI do magnetofonu je použito volných špiček 1 a 4 konektoru LINE. Vlastní obvody podle obr. 1 jsou umístěny uvnitř magnetofonu a jsou připojeny k ovládacím obvodům magnetofonu a zároveň napájeny přes konektorovou zástrčku na desce DO označenou 01. Tento konektor se nachází za mechanikou magnetofonu a čísla špiček na obr. 1 odpovídají jejich pořadí na konektoru, počítáno shora.



Obr. 2.

Na obr. 2 je nakreslena potřebná úprava kabelu KB-05. Připojení magnetofonu je vyřešeno bez zásahu do jeho elektroniky a umožňuje tak bez omezení i jeho běžné používání.

OSMIMÍSTVNÝ ZOBRAZOVAČ SE SÉRIOVÝM STYKEM

Ing. Tomáš Navrátil

Výhody sériového propojení mikropočítače s periférií není třeba zvlášť zdůrazňovat.

Pro převod paralelního tvaru na sériovou posloupnost a naopak se zpravidla používá obvodu typu USART. Protože se však jedná o poměrně drahou součástku VLSI a její použití k danému účelu by bylo určitým přepychem, bylo na straně displeje pro převod použito zapojení sestávající ze dvou IO střední hustoty integrace.

Na straně mikropočítače je převod řešen jednoduchým programem. Tento podprogram převádí obsah akumulátoru na sériovou posloupnost impulsů tak, že jednotlivé bity mají tvar podle obr. 1. Impuls přenášející hodnotu logické nuly má tvar 1, impuls přenášející hodnotu logické jedničky má tvar 2. Při každém volání podprogramu je tedy vysílána sekvence osmi impulsů.

Z důvodů maximální jednoduchosti jsou vypuštěny pomocné bity. Funkci start bitu přebírá první impuls.

Impulsy se vysílají jedním z vývodů brány C obvodu 8255. Pro výkonové zesílení je použito zapojení z obr. 2.

Sériovou linkou jsou impulsy přiváděny na vstup optočlenu (obr. 3). Rezistor R1 slouží k nastavení proudu smyčky (20 mA). Diody D1 chrání proti přepólování a dioda D2 zvyšuje imunitu proti rušení tím, že propustí pouze impulsy o napětí větším než je Zenerovo napětí.

Tranzistor T1 spolu s polovinou IO1 a rezistory R2 až R5 tvoří tvarovací obvod impulsů, které jsou po přechodu linkou zpravidla tvarově deformovány. Upravené impulsy jsou přivedeny na vstup zapojení (obr. 4) pro převod sériové posloupnosti na paralelní tvar (IO2 a IO3) (příhláška o autorské osvědčení PV 6076-85).

Čelem prvního impulsu se spustí monostabilní obvod 2a. Výstup Q obvodu 2a změní úroveň H na L. Tato změna způsobí spuštění monostabilního obvodu 2b, což se projeví změnou na výstupu Q obvodu 2b z úrovně H na L.

Po uplynutí časového intervalu T_{p1} (určen hodnotami C1 a R6) se vrací stav výstupu Q obvodu 2a na úroveň H. Tato změna způsobí posun logických hodnot v posuvném registru IO3 tak, že logické hodnoty na výstupech Q1 až Q8

Tab. 1.

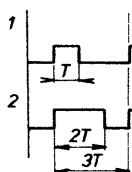
D7	D6	D5	D4	D3	D2	D1	D0
D. t.	Y	Y	Y	X	X	X	X

číslice v kódu BCD

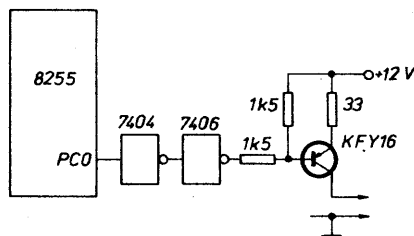
- 000 — 1. pozice zprava
- 001 — 2. pozice zprava
- 010 — 3. pozice zprava
- 011 — 4. pozice zprava
- 100 — 5. pozice zprava
- 101 — 6. pozice zprava
- 110 — 7. pozice zprava
- 111 — 8. pozice zprava

desetinná tečka

- 1 — svítí
- 0 — nesvítí



Obr. 1



Obr. 2

se posunou o jedno místo doprava a současně logická hodnota na vstupu &A se zapíše na výstup Q1. V případě, že pro T_{p1} dodržíme podmínku:

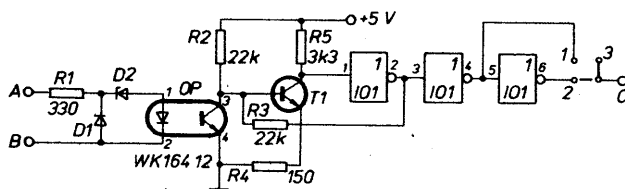
$T \dots$ viz obr. 1

$$t_{p1} = T + T/2 \quad [1]$$

dojde po každém čele vstupního impulsu a po opětovné činnosti IO2a ke vzorkování délky daného impulsu a na výstupu Q1 IO3 se objevují logické nuly a jedničky v závislosti na tom, zda přišel na vstup C impuls průběhu 1 nebo 2 z obr. 1.

Časový interval t_{p2} obvodu IO2b (určen hodnotami C2 a R9) je zvolen tak, aby byl obvod znovu spouštěn. Výstup Q obvodu IO2b přechází tedy po příchodu čela prvního impulsu na úroveň L a zůstává v tomto stavu po dobu vzorkování všech osmi impulsů. Při dodržení podmínky:

$$t_{p2} > 3T \quad [2]$$



Obr. 3.

signalizuje návrat výstupu Q obvodu IO2b z úrovně L na H, že právě došlo k převodu celé sekvence osmi impulsů (obdoba signálu „data platná“).

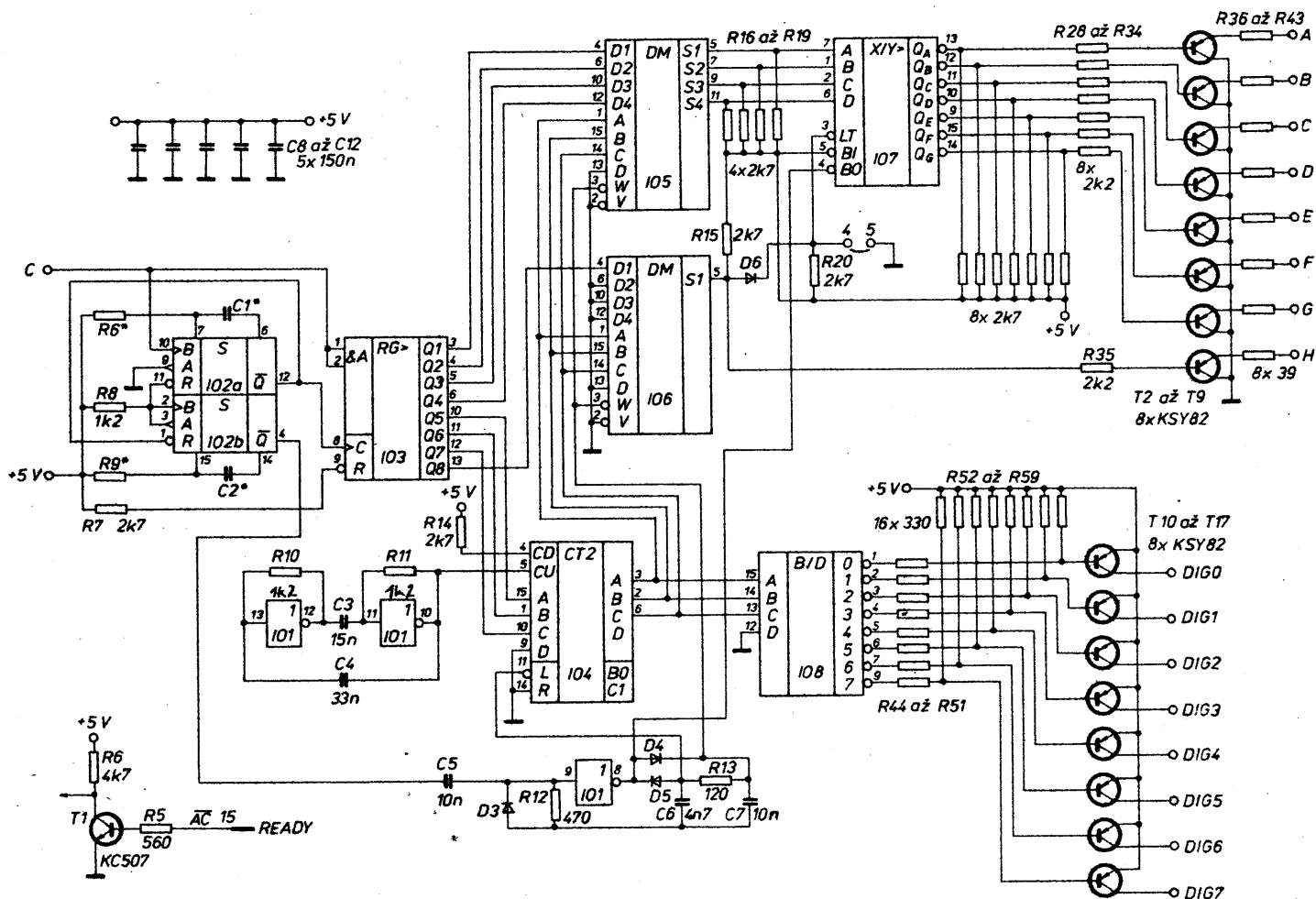
Na výstupech Q1 až Q8 rezistoru IO3 se v tomto okamžiku objeví obsah akumulátoru mikroprocesoru. Pro další použití tohoto bajtu v logice displeje je jednotlivým bitům udělen význam podle tab. 1.

Bity D0 až D3 představují BCD hodnotu zobrazovaného čísla, bity D4 až D6 pozici, na které je toto číslo zobrazováno s případnou desetinnou tečkou (bit B7).

V prvním režimu, to znamená v době, kdy na vstup C nepřichází sériová posloupnost, je na vstupu L čítače IO4 úroveň H. Protože na vstup CU jsou přiváděny impulsy z generátoru tvořeného dvěma invertory (na kmitočtu v širokých mezích nezáleží), objevují se na výstupech A až C kombinace 1 až 8. Tyto kombinace jsou přivedeny na vstupy A až C dekodéru IO8, takže na displeji jsou cyklicky spínány jednotlivé číslovky. Dále jsou stejné kombinace přivedeny na adresové vstupy A až C paměťových obvodů IO5 a IO6. Protože na vstupech W se nachází úroveň H, jsou obě paměti ve stavu čtení a na výstupech S1 až S4 se objevují obsahy jednotlivých paměťových míst, které jsou po dekodování obvodem IO7 cyklicky zobrazovány na odpovídajících místech displeje.

V paměti IO5 jsou uchovány hodnoty osmi číslic, v paměti IO6 hodnoty desetinných teček. Kapacita obou pamětí není příliš ekonomicky využita, jsou to však jediné dostupné paměti s vhodnou organizací pro tento účel.

V druhém režimu se provádí zápis do paměti. Po převodu sekvence, v okamžiku, kdy Q obvodu IO2b přechází z úrovně L do H, se uvede do činnosti obvod skládající se z D3 až 5, C5 až C7, R12, R13 a jednoho invertoru. Derivační člen C5, R12 vygeneruje na vstupu 9 obvodu IO1 kladný impuls. Výstup 8 přejde na okamžik na úroveň L. Prostřednictvím diody D5 se tato úroveň objeví také na vstupu L čítače IO4, což způsobí přepis výstupů Q5 až Q7 obvodu IO3 na výstup čítače IO4 a zaadresování paměťového místa v IO5 a IO6.



Obr. 4. Schéma zobrazovače

Po určitém časovém okamžiku (po ustálení hodnot na výstupech IO4) určeném členem D5, R13, C7 se úroveň L objeví také na vstupech W obvodů IO5, IO6 a tím se do vybraného paměťového místa zapíše stav na vývodech Q1 až Q4 a Q8 obvodu IO3 (BCD hodnota číslice a desetinná tečka). Po odeznění záporného impulsu na výstupu 8 obvodu IO1 se prostřednictvím diody D4 vrátí stav na vstupech W IO5 a IO6 z úrovně L do H a za určitý časový interval určený členem R13, C6 se také stav na vstupu L IO4 vrátí z úrovně L do H.

Dále se obnoví první režim, který trvá až do konce převodu další sekvence. Dioda D3 chrání vstup 9 IO3 proti špičkám záporné polarity. Spojení vývodu 8 IO1 se vstupem B0 obvodu IO7 zajišťuje zhasnutí displeje v době zápisu do paměti.

Pro úplnost je třeba dodat, že použitá paměť (IO5, IO6) poskytuje na výstupu negované hodnoty uchovaných dat. Pro správnou činnost daného zapojení je třeba bity D0 až D3 invertovat. Bylo by možné použít čtyř invertorů, výhodnější je však zařazení instrukce XRI 0FH do dále popsaného podprogramu.

Programové řešení převodu paralelního tvaru na sériovou posloupnost je v Tab. 2. Vstupním parametrem podprogramu je hodnota bajtu v akumulátoru se stanoveným významem jednotlivých bitů. Pomocný parametr RATE určuje délku časové prodlevy TN1.

Pro stanovení celkové doby T je potřeba sečíst doby trvání všech instrukcí počínaje okamžikem nastavení výstupu na úroveň log. 1 do okamžiku nastavení výstupu na úroveň log. 0. V daném případě

$$T = (76 + 15 \cdot \text{RATE}) \cdot t \quad [3],$$

t ... doba periody hodinových pulsů (zpravidla 0,5 μs).

Přenosovou rychlost (dobu T) je možné pomocí parametru RATE v širokých mezích měnit.

Na základě zvolené doby T je nutné určit hodnoty R6, R9, C1, C2.

Pro funkční vzorek byla zvolena hodnota RATE = 32 (20 HEX). Dále byl zvolen odpor rezistorů R6 = 39 000 Ω, R9 = 39 000 Ω. Pomocí vztahu [1], [2] a vztahu

$$C = T / (\text{KONST}1 \cdot R \cdot (1 + 0,7/R)) \quad [4],$$

kde KONST1 = 0,28 pro UC74123 N, = 0,32 pro SN74123 N

byly vypočítány C1 = 33,4 nF, C2 > 66,8 nF. Zvoleny byly C1 = 33 nF, C2 = 0,1 μF.

Tab. 2. Program převodu

COMMENT		LOC	
PROGRAMOVÝ SEGMENT PROVÁDĚJÍCÍ PŘEVOD PARALELNÍHO FORMÁTU NA SÉRIOVOU POSTUPNOST BITŮ. PRO VÝSTUP POSTUPNOSTI JE POUŽITA BRANA C OBVODU 8255. VÝSTUP NA BRANĚ JE OVLADÁN POMOČÍ ŘIDIČI BRANY CNT. PŘEDPOKLÁDÁ SE POČATEČNÍ INICIALIZACE OBVODU 8255 NA REŽIM 0.		LOC1	CALL P1 CALL P3 CALL P2 JMP SKOK
CNT	EGU 7	SKOK	DCR C MOV A,H JNZ CYKL POP H POP D POP B POP PSH EI RET
RATE	EGU 32	P1	MVI A,03H OUT CNT RET
PUBLIC SERIO CSEG D1 PUSH PSH PUSH B PUSH D PUSH H MVI B,RATE		NASTAV VÝSTUP NA LOC 1	
SERIO		CYKL	
MVI C,8		MVI A,03H	
XRI 0FH		OUT CNT	
RLC		RET	
MOV M,A		PAGE 1-1	
JC LOC1		P2	
MVI A,03H		OUT CNT	
OUT CNT		RET	
MVI A,03H		PAGE 1-1	
OUT CNT		P3	
RET		MOV D,B	
PAGE 1-1		DCR D	
P2		JNZ TN1	
P3		RET	
TN1		END	

Seznam součástek

Rezistory

R1	330 Ω (MLT — 0,25)
R6', R9'	39 k Ω , TR 191 (TR 175)
R2, R3	22 k Ω TR212
R4	150 Ω
R5	3,3 k Ω
R7, R14,	
R15 až R27	2,7 k Ω
R8	1,2 k Ω
R10, R11	1,2 k Ω
R12	470 Ω
R13	120 Ω
R28 až R35	2,2 k Ω
R36 až R43	39 Ω (MLT — 0,25)
R44 až R59	330 Ω

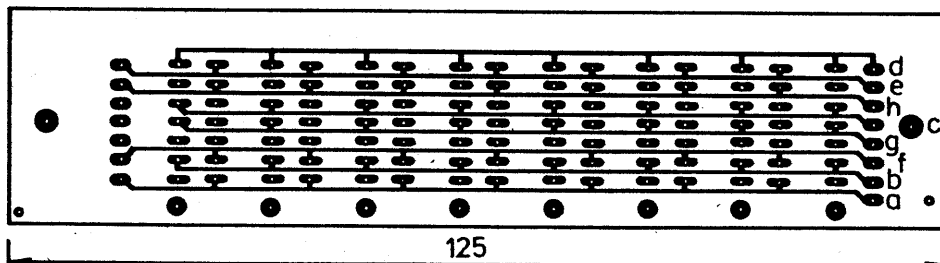
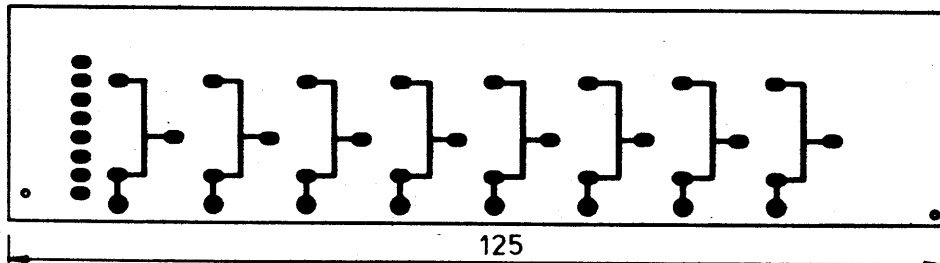
Kondenzátory

C1'	33 nF, TC216
C2'	0,1 μ F, TC2155
C3	15 nF, TK744
C4	33 nF, TK782
C5	10 nF, TK724
C6	4,7 nF, TK724
C7	10 nF, TK724
C8 až C12	0,15 μ F, TK782

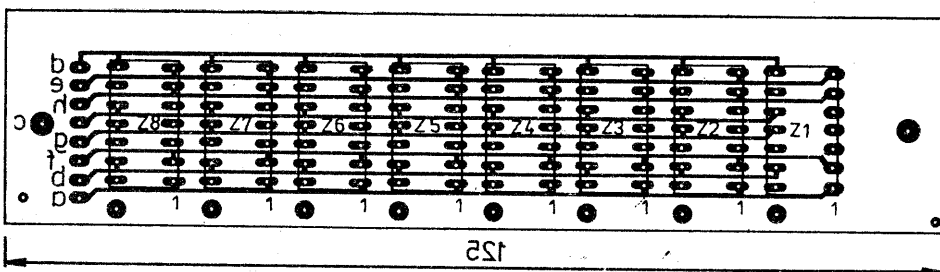
Polovodičové součástky

D1, D3 až D6	KA221
D2	KZ260/6V8
T1	KSY71
T2 až T17	KSY82
IO1	MH7404
IO2	UCY 74123N
IO3	MH74164
IO4	MH74193
IO5, IO6	MH7489
IO7	D146
IO8	MH7442
OP	WK 16412
Z1 až Z8	LQ470

viz text



Obr. 7. Obrázky plošných spojů desky W507 displeje



Obr. 8. Rozmístění součástek na desce s plošnými spoji W507 displeje

Konstrukce

Displej je vestavěn v jednoduché krabici z hliníkového plechu. Síťový transformátor s usměrňovačem jsou umístěny mimo krabici. Na zadní stěně je stabilizátor MA 7805. Pro připojení sériové linky je použit běžný tříkolíkový konektor.

Jako náhrada prokovených děr se mi osvědčil tenký pocínovaný drátek vložený do otvoru a připájený z obou stran ke spoji. Při pájení nesmí do otvoru vtéci kalafuna, aby bylo možno následně zasunout a připájet IO. Tímto způsobem jsou připájeny především číslicovky LQ470 (případně jejich objímky).

Oba oboustranné spoje jsou po osazení přilepeny k sobě, propojeny 16 vodiči a přišroubovány čtyřmi šrouby s distančními podložkami. Protože při případném kontaktu vodičů A až H s kostrou by došlo ke zničení číslicovek, doporučuji příslušná místa izolovat např. přilepením kousku odleptaného kuprexitu.

Obr. 6. Rozmístění součástek na desce s plošnými spoji W506 zobrazovače

Uvedení do chodu

Logickou sondou zkontrolujeme, zda pracuje generátor impulsů (výstup 10 IO1). V prvním režimu musí být na výstupu 8 IO1, vstupech 11 IO4, 3 IO5 a IO6, 4 IO7 úroveň H. Na výstupech A, B, C se objevují impulsy z generátoru dělené 2, 4 a 8. Tyto impulsy zkontrolujeme také na příslušných vstupech IO5, IO6, IO8. Dále zkontrolujeme, zda jsou tranzistory T10 až T17 cyklicky spínány, přičemž propojením bodů 4 a 5 se musí rozsvítit všechny segmenty.

Další krok již vyžaduje spojení s mikropočítačem. Na obr. 2 je uvedena pouze jedna z možností výkonového zesílení signálu a napájení proudové smyčky. Je možné použít přímo výstup některého z výkonových hradel TTL nebo výstup obvodu 3212 (v tomto případě by bylo třeba nahradit diodu D2 z obr. 3 drátovou propojkou).

Pro případ, že obvod napájející proudovou smyčku signál invertuje, jsou určeny přepínací body 1—3, 2—3. Propojku zvolíme tak, aby v době, kdy nepřichází žádná posloupnost, byla na vstupech 1, 2 obvodu IO3 úroveň L.

Nyní do akumulátoru vložíme určitou hodnotu a vyvoláme podprogram. Při správné činnosti se na výstupech Q1 až Q8 IO3 musí objevit obsah akumulátoru (Q1 až Q4 obsahují negované úrovně) a na displeji se rozsvítí příslušná číslice.

Princip kódování, který je zde použit, zaručuje velkou odolnost proti kolísání přenosové rychlosti (můžeme se o tom přesvědčit měněním parametru RATE v okolí zvolené hodnoty), přesto doporučuji z hlediska dlouhodobé stability na místech R6, R9, C1, C2 použít řadu TR191 a TC215.

Použití

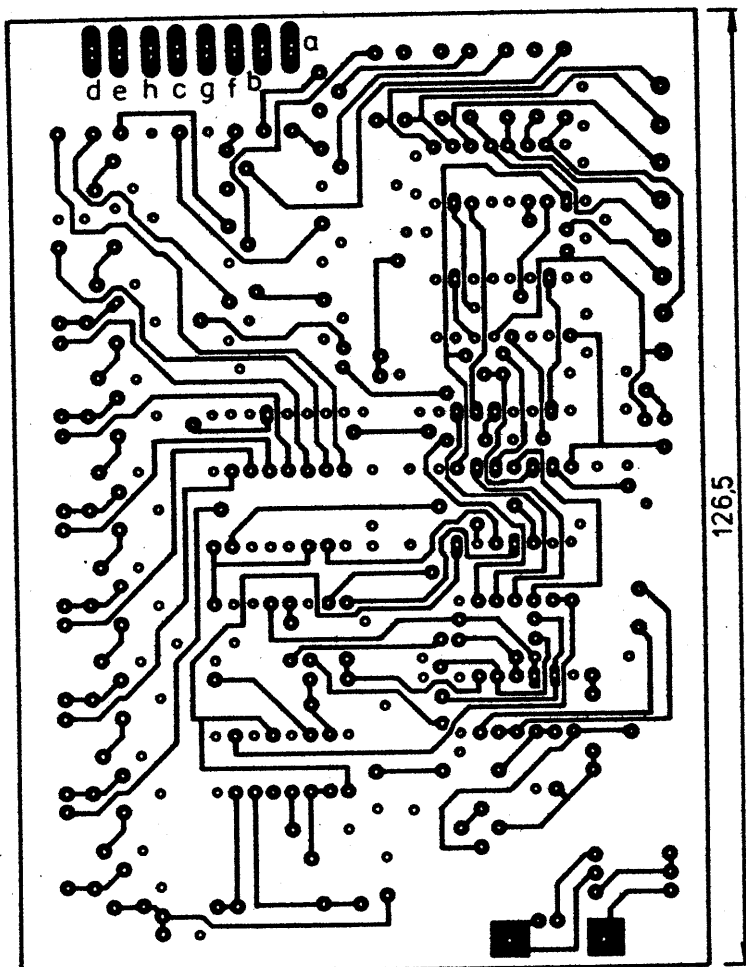
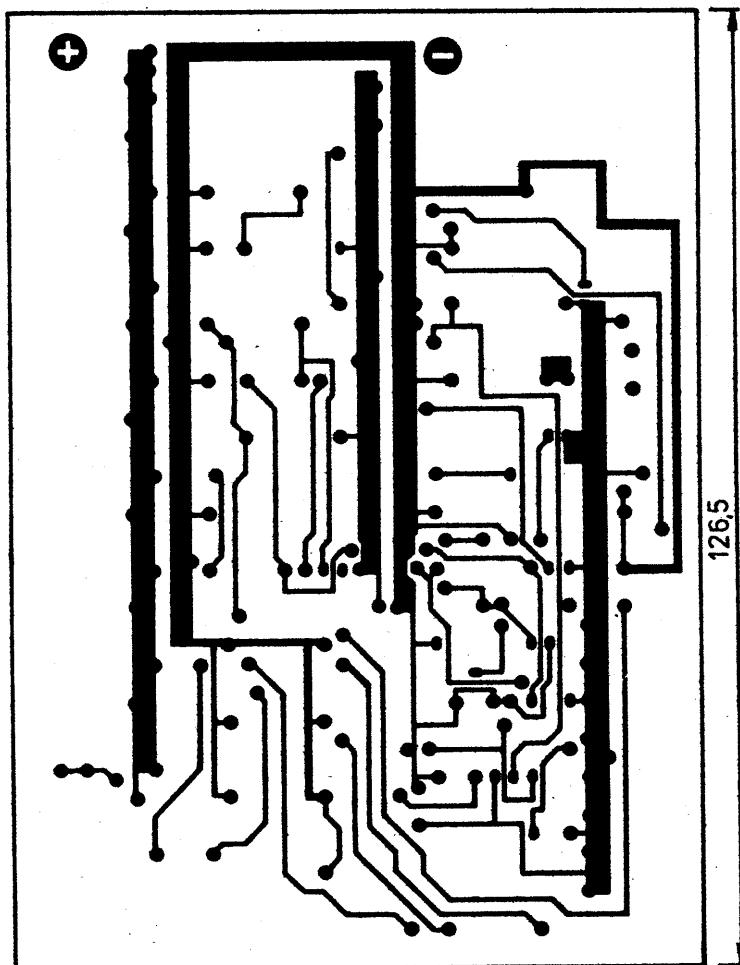
Displej najde uplatnění všude tam, kde je potřebné zobrazit výsledek nějaké operace (např. měření elektrických veličin, zobrazení reálného času apod.), přičemž použití terminálu se jeví pro daný účel jako nevhodné.

Displej je sice určen pro zobrazení desítkových čísel, ale není vyloučeno zobrazení čísel hexadecimálních. V tomto případě by bylo výhodnější místo obvodu D146 použít obvod U40511 s příslušnými úpravami. Provedení s D146 však poskytuje možnost programového zhasínání jednotlivých číslic (jedničkové bity D0 až D3).

Použití zapojení z obr. 3 zajišťuje galvanické oddělení a umožňuje přenášet číselný údaj na poměrně velkou vzdálenost (stovky metrů).

V případě, že zobrazovač bude umístěn v blízkosti mikropočítače, je možné toto zapojení vypustit a signál přivést přímo na svorku C (obr. 4).

Obr. 5. Obrázce plošných spojů desky W506 zobrazovače



Mikropočítač kompatibilní se ZX-Spectrum

Typ programu PROM 16Kb
Kč 10 000, 4 Kč 10

0000	AF	00100	ORG	00H	0070	13	00720	INC	DE
0001	D300	00120	START	XOR A	0071	1802	00730	JR	DEC
0003	31F0FF	00130	OUT	(0), A	0073	D023	00740	NEXT	INC
0006	DD210000	00140	LD	SP, 0FFF0H	0075	1B	00750	DEC	DEC
000A	110040	00150	LD	IX, 8000H	0076	08	00760	EX	AF, AF'
000D	CD1A00	00160	LD	DE, 4000H	0077	06B2	00770	LD	B, 0B2H
0010	3004	00170	CALL	BYTES	0079	2E01	00780	MARKER	LD
0012	D801	00180	JR	NC, ERR	007B	CD9700	00790	BIT58	CALL
0014	00	00190	OUT	(1), A	007E	D0	00800	RET	NC
0015	00	00200	NOP		007F	3EC8	00810	LD	A, 0C8H
0016	D302	00210	NOP		0081	B8	00820	CP	B
0018	10E6	00220	ERR	OUT	(2), A	0082	CB15	00830	RL
001A	14	00230	JR	START		0084	06B0	00840	LD
001B	08	00240	BYTES	INC	D	0086	D27B00	00850	JP
001C	15	00250	EX	AF, AF'		0089	7C	00860	LD
001D	3E0F	00260	DEC	D		008A	AD	00870	XOR
001F	D3FE	00270	LD	A, 0FH		0088	67	00880	LD
0021	0E02	00280	OUT	(0FEH), A		008C	7A	00890	LD
0023	CD9B00	00290	LD	C, 02H		008D	B3	00900	OR
0026	30FB	00300	LDSTRT	CALL	EDGE1	008E	2000	00910	JR
0028	211504	00310	JR	NC, LDSTRT		0090	7C	00920	LD
002B	10FE	00320	LD	HL, 0415H		0091	FE01	00930	CP
002D	2B	00330	WAIT	DJNZ	WAIT	0093	D20000	00940	JP
002E	7C	00340	DEC	HL		0096	C9	00950	RET
002F	B5	00350	LD	A, H		0097	CD9B00	00960	EDGE2
0030	20F9	00360	OR	L		009A	D0	00970	CALL
0032	CD9700	00370	JR	NZ, WAIT		009B	3E16	00980	EDGE1
0035	30EC	00380	CALL	EDGE2		009D	3D	00990	DELAY
0037	069C	00390	JR	NC, LDSTRT		009E	20FD	01000	JR
0039	CD9700	00400	LEADER	LD	B, 9CH	00A0	A7	01010	AND
003C	30E5	00410	CALL	EDGE2		00A1	04	01020	SAMPLE
003E	3EC6	00420	JR	NC, LDSTRT		00A2	C8	01030	INC
0040	B8	00430	LD	A, 0C6H		00A3	3E7F	01040	LD
0041	30E0	00440	CP	B		00A5	D8FE	01050	IN
0043	24	00450	JR	NC, LDSTRT		00A7	1F	01060	RRR
0044	20F1	00460	INC	H		00A8	D0	01070	RET
0046	06C9	00470	JR	NZ, LEADER		00A9	A9	01080	XOR
0048	CD9B00	00480	LD	B, 0C9H		00AA	E620	01090	AND
004B	30D6	00490	CALL	EDGE1		00AC	28F3	01100	JR
004D	78	00500	JR	NC, LDSTRT		00AE	79	01110	LD
004E	FED4	00510	LD	A, B		00AF	2F	01120	CPL
0050	30F4	00520	CP	0D4H		00B0	4F	01130	LD
0052	CD9B00	00530	JR	NC, SYNC		00B1	E607	01140	AND
0055	D0	00540	CALL	EDGE1		00B3	F608	01150	OR
0056	79	00550	RET	NC		00B5	D3FE	01160	OUT
0057	EE03	00560	LD	A, C		00B7	37	01170	SCF
0059	4F	00570	XOR	03H		00B8	C9	01180	RET
005A	2600	00580	LD	C, A		00B0		01190	END
005C	06B0	00590	LD	H, 0		00000	TOTAL ERRORS		START
005E	1819	00600	LD	B, 0B0H		BIT58	007B		
0060	08	00610	JR	MARKER		BYTES	001A		MARKER
0061	2005	00620	EX	AF, AF'		DEC	0075		NEXT
0063	DD7500	00630	JR	NZ, FLAG		DELAY	0090		SAMPLE
0066	180B	00640	LD	(IX+0), L		EDGE1	009B		START
0068	CB11	00650	JR	NEXT		EDGE2	0097		SYNC
006A	2C	00660	RL	C		ERR	0016		WAIT
006B	2093	00670	INC	L		FLAG	0068		
006D	79	00680	JR	NZ, START		LDSTRT	0023		
006E	1F	00690	LD	A, C		LEADER	0037		
006F	4F	00700	RRR			LOOP	0060		
		00710	LD	C, A					

Obsah pouzdra 74S287 připojovaného na D4 až D7

.D100 1FF
0100 FF F3 F0 F1 F0 FF FD F1 F0 F0 F1 F0 F0 FD F8 F0
0110 F0 F2 F3 F0 F3 F0 F8 F8 F4 F8 F5 FE FF F3 FE FE
0120 F2 F0 F9 F0 F0 F8 F1 F5 F2 F0 FE F8 FC F5 F0 F9
0130 FD F5 F0 F0 FC F6 FC FD F3 F0 F0 F5 FE F6 F8 F0
0140 F0 F4 F0 F1 F6 F9 FD F9 F0 F0 F6 F8 FE F4 F0 F4
0150 FD F9 F0 F0 F9 FE F3 FF F6 F0 F6 F0 F8 F9 F8 F0
0160 F5 FD F5 F0 F8 F8 FB F1 FC F0 F5 F9 FF FF F3 F8
0170 F2 FD F3 FB F8 F6 F2 FE F1 FD F5 F0 F0 FE F8 F8
0180 FB F5 F6 F0 F2 F9 F0 FC FD F7 FA F3 F0 F0 FC FE
0190 F1 F2 F0 F0 F9 FD F9 F0 F0 FE F6 FD F0 FD F7 F4
01A0 F8 FE FF FB FE FF F0 F9 F6 F0 F8 F3 F9 FF FF F4
01B0 F7 F6 F8 F3 FE F7 F9 F0 F0 F0 F0 F0 F0 F0 F0
01C0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0
01D0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0
01E0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0
01F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0

Obsah pouzdra 74S287 připojovaného na D0 až D3

.D200 2FF
0200 FA FD F0 F3 FF FF FD F2 F0 F8 F1 F0 F4 FC F1 F0
0210 F3 F0 FD F0 FD F0 F1 FE F1 F0 F1 F3 F0 FD FF F0
0220 F0 FC F9 F0 F3 FF F2 F1 F0 F1 FF F2 F7 FB F2 FF
0230 FC F9 F0 F3 FE F0 F9 FC F9 F0 F3 F0 FC FB F3
0240 FE F2 F2 FF F0 FC FC F9 F0 F3 FD F7 FF FD F3 FF
0250 FC F9 F0 FD F7 FE F0 F4 F2 F0 F0 FB F1 F1 F0 F2
0260 F0 FD F7 F0 F1 F0 FC F1 F2 F2 F9 F7 F1 F4 F1 F1
0270 F0 FD F2 F1 F0 F0 FB F2 F0 FC F9 F0 FD F3 FC FB
0280 FC F1 F0 FB FD F7 F0 F7 FA F6 F7 FB F2 FD F7 FF
0290 F0 FD F0 F0 FC FC F9 F0 FD F3 F1 F3 F2 FF FA F0
02A0 FC F3 F7 FD FF F1 FD FA FE F2 F2 FF F2 F4 FE
02B0 F0 FF F0 FD FF F3 FC F0 F0 F0 F0 F0 F0 F0 F0
02C0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0
02D0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0
02E0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0
02F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0

PROFESOR

ing. Martin Štěpánek

Jordana Jovkova 3256
Praha 4-Modřany

Program PROFESOR je určen pro zkoušení a opakování látky různých oborů zábavnou formou. Je zvláště vhodný pro problematiku, kde záleží na grafickém zobrazení (např. zeměpis, astronomie, geometrie, elektronika apod.). Program je ovšem použitelný i pro zcela negrafické obory jako je třeba mluvnice nebo zkoušení násobilky. Program PROFESOR pracuje v několika režimech, v nichž můžeme látku opakovat (popř. vykládat), zkoušet a hodnotit. Zkoušení může probíhat buď individuálně, nebo soutěží více účastníků mezi sebou.

Program PROFESOR sám o sobě zajišťuje komunikaci mezi počítačem a člověkem. Aby PROFESOR ožil, musí obsahovat databázi údajů z oboru, který chceme zkoušet. Takových datových bází může být velké množství a vždy, když nahrajeme novou databázi do programu PROFESOR, získáme znalce příslušného oboru.

Aby se snadno mohly vytvářet nové databáze podle zájmů jednotlivých uživatelů, přísluší k programu PROFESOR ještě pomůcka zvaná TESTEDITOR. TESTEDITOR je program, který velmi snadno umožňuje připravovat a opravovat báze grafických a textových informací vlastních testů (viz str. 34).

Program PROFESOR byl sestaven tak, aby byl snadno přenositelný na jiné typy počítačů. Pro používání programu PROFESOR na jiném počítači se upraví jen vlastní program a všechny databáze je možné používat beze změn.

Obsluha programu PROFESOR

Po nahrání programu do počítače se program sám spustí a my se dostaneme do jeho hlavního menu. V horní části hlavního menu vidíme tři možné režimy činnosti programu PROFESOR. Režim výkladu („v“) slouží k postupnému prohlížení obsahu celé databáze. Vykreslí se obrázek jedné otázky a vypíše se také správná odpověď. Například se zobrazí poloha města v obrysu Evropy (otázka) a jeho název (odpověď). U některých databází se také zobrazí nabízené varianty odpovědí, ze kterých bychom si v režimu zkoušení vybírali. Po prohlédnutí otázky můžeme buď přejít k další otázce klávesou „ENTER“, nebo režim výkladu ukončit klávesou „SPACE“.

Dalším režimem je zkoušení. V tomto režimu se nejprve nakreslí podkladový (statický) obrázek. Poté jsme dotázáni na počet účastníků, jejich jména, počet otázek v kole a stupeň obtížnosti. Volba stupně obtížnosti se projevuje v čase poskytnutém na zodpovězení otázky. Základní čas (obtížnost jedna) je ovšem různý pro každou databázi (podle její problematiky). Po volbě obtížnosti započne vlastní proces zkoušení. U jména účastníka, který má odpovídat,

se rozsvítí blikající kurzor „“, kterým můžeme cyklicky pohybovat pomocí klávesy „SPACE“ (mezerník). Když se domníváme, že je výběrový kurzor před správnou odpovědí, stiskneme „ENTER“. Zvukový signál nás upozorní na úspěšnost naší odpovědi. Tři stoupající tóny nám oznamují správnou odpověď, jeden dlouhý a hluboký tón odpověď špatnou. Při volbě správné odpovědi začne vedle našeho jména narůstat pás značek, který svou délkou vyjadřuje okamžitou úspěšnost. V každém případě se závěrem otázky zobrazí správná odpověď. Víme již, že čekání na odpověď je časově omezeno. Abychom mohli posoudit zbývající čas, zobrazuje se pod obrázkem souvislý pás, který se s ubývajícím časem zkracuje. V tomto páse je také informace o pořadí otázky. Po vyčerpání všech otázek dochází k vyhodnocení, které je slovní a procentuální. Po stisku klávesy se vracíme do hlavního menu. Příkazem nahrání nové databáze („n“) můžeme změnit obor znalostí našeho počítače. V režimu nahrávání uvedeme nejprve jméno databáze a poté určíme druh paměťového zařízení. Po ukončení nahrávání se vrátíme do hlavního menu.

Ve spodní části hlavního menu můžete vidět základní informace o databázi právě obsažené v programu PROFESOR. Je to jednak tabulka způsobu hodnocení v závislosti na úspěšnosti odpovědi a dále název datové báze, počet obrázků v ní (jednička představuje trvalý podkladový obrázek) a za lomítkem si přečteme kolik buněk vymezeného prostoru zabírá (u první verze programu PROFESOR je to maximálně 4000 buněk).

Princip činnosti programu

Program PROFESOR pracuje s jedním statickým obrázkem a sadou obrázků dynamických, které se překreslují přes statický podklad. Každý dynamický obrázek v podstatě přísluší jedné otázce testu. Při výuce se otázky zobrazují ve stejném pořadí, jak jsou uloženy v databázi. Při zkoušení je pořadí náhodné. Každý obrázek je popsán jako série převážně grafických operací ve speciálním jazyce. Program PROFESOR pak vystupuje v roli interpretačního překladače. Zde je třeba zdůraznit, že pro užívání programu PROFESOR a TESTEDITOR není vůbec nutné znát zmiňovaný vnitřní jazyk. Program TESTEDITOR například píše

program v tomto jazyce sám, podle toho jak kreslíme kurzorem po obrazovce.

Přenosnost programu PROFESOR na jiné počítače

Program PROFESOR je navržen tak, aby jeho úpravy při přenosu byly minimální. Od cílového počítače ovšem vyžaduje splnění některých základních vlastností. Protože je PROFESOR programem grafickým, musí mít i cílový počítač bodovou grafiku. Protože se jedná o kombinaci obrázků s textem, musí být také zachovány následující proporce. Program bude pracovat na grafické ploše 256X192 bodů (může být i fiktivní). Pro texty bude tato plocha odpovídat 24 řádkům po 32 znacích. Tyto relace jsou splněny u celé řady počítačů přímo (kromě Spectra např. počítače značky Commodore, Atari, Amstrad-Schneider a typy kompatibilní s IBM PC nebo u nás známý Sord M5). U počítače PMD-85, který nespĺňuje uvedený formát, lze s výhodou použít například příkaz SCALE (PROFESOR na PMD-85 již existuje). Cílový počítač také musí disponovat operační pamětí alespoň 16 kB. I když jsou všechny databáze barevné, není nutné je provozovat na barevném počítači. Všeobecně lze říci, že obslužné podprogramy funkcí, které nelze na cílovém počítači realizovat (např. barvy, zvuk apod.), musíme zachovat, pouze v nich neuvedeme výkonnou instrukci, kterou počítač neumí. Pokud je program PROFESOR správně převeden, budou databáze fungovat beze všech změn.

Nyní bych se ještě zmínil o některých příkazech jazyka BASIC počítače ZX-Spectrum, které by mohly činit potíže při přenosu programu. Příkaz RANDOMIZE (ř. 1180) nastavuje semeno pseudonáhodných čísel podle vnitřního čítače času (tedy náhodně). Funkce RND generuje pseudonáhodná čísla v intervalu od nuly do jedné. Příkaz DRAW je relativní. Grafické souřadnice se ve směru „x“ počítají od 0 do 255 zleva doprava a ve směru „y“ od 0 do 175 zdola nahoru. Řádky textu se značí 0 až 21 shora dolů a sloupce 0 až 31 zleva doprava. Pro tisk na řádek 22 (leží bezprostředně pod linií y=0) nebo 23 užíváme příkaz PRINT # 0; AT 0, 0 nebo PRINT # 0; AT 1,0. Příkaz INVERSE slouží jen pro grafické zvýraznění a lze ho vynechat. Příkaz OVER je využíván k mazání. Pro OVER 0 se vynášejí body systémem „or“ pro OVER 1 systémem „xor“. Pro všechny základní báze dat může grafika pracovat neustále systémem „xor“ nebo lze nahrazovat příkaz OVER 1 režimem „reset“, kdy se pod vynesenu pozici vždy nuluje. V databázích, u kterých by se s touto eventualitou nepočítalo, by docházelo jen k malým grafickým vadám (např. v místě křížení by nebyl vyneseno bod apod.). Základní databáze jsou ovšem navrženy tak, aby se předešlo zmiňovaným a některým dalším problémům. Program také využívá jeden uživatelem definovaný znak (ř. 1930 a ř. 2830), který může být samozřejmě nahrazen znakem běžným. Příkaz BORDER slouží k nastavení barvy okraje a lze ho vynechat. Klávesa „ENTER“ generuje kód 13, na který je testována. Generuje-li odesílací klávesa cílového počítače jiný kód, musí se testovací místa přizpůsobit.

Základní databáze programu „Profesor“

V této kapitole uvádím jako příklad základní vlastnosti některých databází, které mám pro program PROFESOR sestavené.

Dopravní situace

Návěští „DopravSit“, 27+1 obrázek, délka 3992 buněk. Databáze se zabývá řešením dopravních situací na křižovatkách.

DOPRAVNÍ 1986 SITUACE c. M. Štěpánek		
Které pojede poslední?		
a) 1 b) 2 c) 3		
Správně: Pavel Martin Petr		

Evropská moře

Návěští „MoreEvr“, 16+1 obrázek, délka 2116 buněk. Zkoušený zde vybírá jeden z pěti názvů moří, o kterém se domnívá, že přísluší vyšrafované oblasti na mapě Evropy.

	EVROPSKÁ 1986 MOŘE M. Štěpánek
	Baltické Barentsovo Tyrhénské Atlantické Severní
	Správná odpověď: Atlantik
	Pavel Milan Josef

Malá násobilka

Návěští „Násobilka“, 31+1 obrázek, délka 2754 buněk. Program zkouší malou násobilku formou nabídky příkladu a pěti možných odpovědí. Oba činitele jsou menší než deset.

MALÁ NÁSOBILKA c. Martin Štěpánek 1986	
Kolik je: 5 x 7 =	
a) 48 b) 350 c) 35 d) 3500 e) 35000	7 4 8 0 0 0 5 7 ----- 4 9 0 0 0
Pavel Petr Michal Jana	

Evropská města

Návěští „MestaEvr“, 35+1 obrázek, délka 2373 buněk. Vybíráme jednu z pěti nabízených odpovědí pro polohu města označenou křížkem a šipkou.

	EVROPSKÁ MĚSTA Štěpánek '86
	Napoli Dona München Lisboa Bruxelles
	Správná odpověď: München
	Michal Rene Martin

Souhvězdí severní oblohy

Návěští „Souhvězdí“, 20+1 obrázek, délka 1928 buněk. Program vykresluje bodově nejznámější souhvězdí severní oblohy. U každého souhvězdí udává velikost desetistupňového úhlu. K výběru je nabízeno pět odpovědí.

* SOUHVĚZDÍ SEVERNÍ OBLOHY *	
	Pegas Orion Blíženci Kasiopeja Velký vůz
	M. Štěpánek c. 1986
Martin Zvonka	

Velká násobilka

Návěští „VelNasI“, 40+1 obrázek, délka 2977 buněk. Zkoušení násobilky, kde první činitel je menší a druhý větší než deset. Správnou odpověď vybíráme z pěti čísel.

* VELKÁ NÁSOBILKA I *	
4x15 7 2 0 3 0 0 4 3 0 5 0 0	2 60 HŠ '86
Jana Iva	

Evropská pohorí

Návěští „PohoriEvr“, 24+1 obrázek, délka 2049 buněk. V mapě Evropy se vykreslí poloha pohorí a na zkoušeném je vybrat správný název z pěti nabízených.

	EVROPSKÁ 1986 POHORÍ M. Štěpánek
	Povolská vr. Fr. středohorí Sierra Morena Kastilské p. Penninu
	Správná odpověď:
	Martinec Zvanovec Hubalek

Světová moře a oceány

Návěští „MoreSveta“, 20+1 obrázek, délka 2030 buněk. Na stínítku se zobrazí obrysy světadílů a šipkami jsou označována jednotlivá moře. Odpověď vybíráme z pěti názvů.

L. Novak M. Šopř	
	MOŘE + OCEÁNY Štěpánek c. 1986
Středozemní moře Jeddellovo moře Východočínské moře Jihčínské moře Beringovo moře	

Velká násobilka II

Návěští „VelNas2“, 25+1 obrázek, délka 2402 buněk. Součin, na jehož výsledek se program dotazuje, je tvořen dvěma čísly z intervalu deset až patnáct. Výsledek vybíráme z pěti odpovědí.

*** VELKÁ NÁSOBILKA II *** c. Martin Štěpánek 1986	
Kolik je: 15 x 13 =	195
a) 205 b) 195 c) 180 d) 175 e) 165	15 13 7 5 15 1 9 5 ----- 1 9 5
Lanka Filip Blanka Tomas	

Města v ČSSR

Návěští „MestaCSSR“, 32+1 obrázek, délka 1695 buněk. Do obrysu naší republiky se zakreslují polohy měst a k výběru je nabízeno šest možností.

* Města v ČSSR *	Město:
	Kladno Supperk Přerov Plzeň Ostrava Supperk
c. Štěpánek	
Houčka Jurkas	


Naše města

Návěští „NaseMesta“, 42+1 obrázek, délka 1301 buněk. K naznačené poloze města vybíráme jméno z nabízených sedmi možností. Výpis této databáze je na str. 33.

```

1 GO TO 1750: REM Studenty sta
2 GO TO 1930: REM Morky start
3 REM Kresleni
4 LET U=P(0)
5 LET U0=U+255*CODE b$(U)+COD
6 b$(U+1)+1
7 LET U=U+2
8 IF U>255 AND 0<>1 THEN LET f
9 f=1
10 IF U>255 THEN LET f=0: RETUR
11 N
12 LET k=CODE b$(U)
13 LET U=U+1
14 IF k=2 THEN GO SUB 250: GO
15 TO 60
16 IF k=1 THEN GO SUB 320: GO
17 TO 60
18 IF k=0 THEN GO SUB 1000: GO
19 TO 60
20 IF k=3 THEN GO SUB 300: GO
21 TO 60
22 IF k=4 THEN GO SUB 420: GO
23 TO 60
24 IF k=9 THEN GO SUB 560: GO
25 TO 60
26 IF k=10 THEN GO SUB 620: GO
27 TO 60
28 IF k=5 THEN GO SUB 480: GO
29 TO 60
30 IF k=6 THEN GO SUB 520: GO
31 TO 60
32 IF k=7 THEN CLS
33 IF k=8 THEN BEEP 0.1,30
34 IF k=11 THEN GO SUB 900
35 IF k=12 THEN GO SUB 940
36 IF k=13 THEN GO SUB 1040
37 GO TO 60
38 REM DRAW
39 DRAW CODE b$(U)-xs, CODE b$(
40 U+1)-ys
41 LET xs=CODE b$(U)
42 LET ys=CODE b$(U+1)
43 LET U=U+2
44 RETURN
45 REM PLOT
46 PLOT CODE b$(U), CODE b$(U+1
47 )
48 LET xs=CODE b$(U)
49 LET ys=CODE b$(U+1)
50 LET U=U+2
51 REM AT
52 PRINT AT CODE b$(U), CODE b$(
53 U+1);
54 LET U=U+2
55 RETURN
56 FOR i=u+1 TO u+CODE b$(U)
57 PRINT b$(i);
58 NEXT i
59 LET U=U+CODE b$(U)+1
60 RETURN
61 REM INK
62 INK CODE b$(U)
63 LET U=U+1

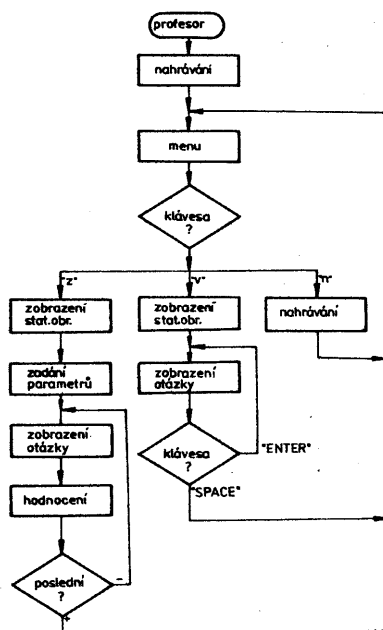
```

	Správný název:
	Volte odpověď:
	Endymion Archimedes M Nectaris Plato M Cognitum M NUBIUM
	MESIC c M. Štěpánek '88 David ♦♦ Helena ♦ Jachym ♦

```

500 RETURN
510 REM PAPER
520 PAPER CODE b$(U)
530 LET U=U+1
540 RETURN
550 REM REL DRAW
560 DRAW CODE b$(U)-128, CODE b$(
57 U+1)-128
58 LET xs=xs+CODE b$(U)-128
59 LET ys=ys+CODE b$(U+1)-128
60 LET U=U+2
61 RETURN
62 REM SEGMENT
63 LET U=0
64 LET U3=U
65 LET U4=U
66 LET U=CODE b$(U)+80
67 GO SUB 20
68 LET U=U+5
69 LET U=U+3
70 LET U=U+4
71 LET U=U+1
72 RETURN
73 REM ERASE
74 LET U=0
75 LET U3=U
76 IF f=1 OR f1=0 THEN RETURN
77 LET U=U+5
78 LET U3=U
79 LET U4=U

```



```

700 LET o=00
800 LET f=1
910 OVER 1
920 GO SUB 20
930 OVER 0
940 LET o=U5
950 LET U=U3
960 LET U=U4
970 LET f=0
980 RETURN
990 REM BORDER
1000 BORDER CODE b$(U)
1010 LET U=U+1
1020 RETURN
1030 REM BLINK
1040 OVER 1
1050 IF f=0 THEN LET f=1: LET
106 no=CODE b$(U)
107 IF no=0 THEN LET f=0: OVER
108 0: LET U=U+1: RETURN
109 LET no=no-1
110 LET U=P(0)
111 GO SUB 20
1120 LET U=P(0)+CODE b$(p(0))
1130 LET U=U+1
1140 RETURN
1150 REM Správná odpověď
1160 LET p=CODE b$(U)
1170 LET U=U+1
1180 RETURN
1190 REM NABÍDKA
1200 GO SUB 20
1210 RANDOMIZE: REM Semeno
1220 INPUT "Pocet účastníků (max
123 (n) )="; h0
1240 IF h0>5 THEN GO TO 1190
1250 FOR h=1 TO h0
1260 INPUT "Jméno "; (h); ". účast
127 nika="; k$
1280 PRINT PAPER 6; INK 8; AT r0+
129 h-1, s0+1; k$
1300 NEXT h
1310 INPUT "Otázka v kole (6-12)
132 ="; p1
1330 IF p1<6 OR p1>12 THEN GO TO
134 1250
1350 INPUT "Oblíbenost 1-3 (1=mal
136 a)"; e
1370 IF e<1 OR e>3 THEN GO TO 12
138 70
1390 LET f2=0
1400 LET f3=0
1410 FOR a=1 TO p1
1420 FOR h=1 TO h0
1430 IF a>1 OR h>2 THEN GO SUB
144 2230
1450 PRINT #0; AT 1,0; "
146 1330 PRINT #0; AT 1,0; INVERSE 1;
147 "Otázka: "; a
1480 PRINT AT r0+h-1, s0; PAPER 6
149 ; INK 8; "
1500 LET o=INT ((p0-1.0001)*RND)
151 +2
1520 IF q=1 THEN GO TO 1450
1530 LET p=INT ((n-0.0001)*RND)+
154 1

```

```

1350 FOR j=1 TO n
1360 IF j=p THEN GO TO 2430
1400 LET U=INT ((p0-1.0001)*RND)
1410 IF U=0 THEN GO TO 1400
1420 LET U=P(0)+3
1430 GO SUB 1080
1440 NEXT j
1450 LET f2=1
1460 LET f3=0
1470 GO SUB 20
1480 LET f5=1
1490 LET f2=0
1500 LET k1=1
1510 PRINT AT r, s-1; "
1520 LET q=INT ((s2/4)
1530 FOR i=1 TO 16
1540 PRINT #0; AT 1,32-j-j; "
1550 FOR i=1 TO 9
1560 LET k$=INKEY$
1570 IF k$=CHR$ 13 THEN LET i=32
1580 LET j=17
1590 IF k$=" " THEN GO SUB 2580
1600 NEXT i
1610 PRINT AT r+k1-1, s-1; "
1620 IF k1=p AND i>=32000 THEN G
1630 O SUB 2770: GO TO 1640
1640 BEEP 1, -6
1650 GO SUB 2460
1660 PRINT AT r0+h-1, s0; PAPER 6
1670 ; INK 8; "
1680 NEXT h
1690 NEXT a
1700 GO SUB 2230
1710 BEEP 0.1,30
1720 PRINT #0; AT 1,0; " *** ZKOU
1730 SENI JE UKONČENO *** "
1740 GO SUB 2460
1750 BEEP 0.1,30
1760 GO SUB 2660
1770 RETURN
1780 PAPER 1
1790 INK 7
1800 BORDER 1
1810 CLS
1820 DIM x(13)
1830 DIM o(5)
1840 DIM p(100)
1850 DIM b$(4000)
1860 LET f=0
1870 LET o=0
1880 LET f2=0
1890 LET f3=0
1900 LET h=1
1910 LET xs=0
1920 LET ys=0
1930 LET f7=0
1940 GO SUB 3220
1950 REM Def znacky (neni nutne)
1960 POKE USR "H"+0, BIN 00000000
1970 POKE USR "H"+1, BIN 00011000
1980 POKE USR "H"+2, BIN 00111100
1990 POKE USR "H"+3, BIN 01111110
2000 POKE USR "H"+4, BIN 01111111
2010 POKE USR "H"+5, BIN 00011100
2020 POKE USR "H"+6, BIN 00011000
2030 POKE USR "H"+7, BIN 00000000
2040 GO TO 2660

```

1. Novotný 2. Mach 3. Štěpánek 4. Hopecský	♦♦ ♦
My jsme chtěli jít dnes do ki- na, ale lístky byly vyprodány.	
Podstatné jm. p (dávne) jm. zámeno číslovka sloveso přídavné přídložka spojka citoslovce částice	Správná odpověď: SLOVNÍ DRUH Štěpánek '88

```

2030 REM Vykład
2040 LET o=1
2050 GO SUB 20
2060 PRINT #0; "-----prerušuje,
2070 pokračuje"
2080 FOR o=2 TO p0
2090 LET j=1
2100 LET f3=1
2110 LET f2=1
2120 GO SUB 20
2130 IF INKEY$=" " THEN GO TO 213
2140 BEEP 0.1,30
2150 IF INKEY$=" " THEN LET o=p0
2160 GO TO 2280
2170 LET f2=1
2180 LET f7=1
2190 GO SUB 2230
2200 LET f2=0
2210 NEXT o
2220 RETURN
2230 REM Vyraz
2240 LET o=0
2250 LET f2=1
2260 LET f3=1
2270 LET U=P(0)+3
2280 IF f7=1 THEN GO TO 2380
2290 GO SUB 1080
2300 GO SUB 2460
2310 LET f3=0
2320 IF q=1 THEN GO TO 2380
2330 FOR i=1 TO n

```

[illegible]

... zbytek všude 32 /mezera/

TESTEDITOR

Ing. Martin Štěpánek

Jordana Jovkova 3256
Praha 4-Modřany

Program TESTEDITOR je pomůcka pro vytváření databází vlastních testů. Editor je graficky orientován, uživatel maluje a popisuje obrázek a databáze se paralelně vytváří sama. Pomocí vestavěného monitoru může zasvěcený uživatel pracovat i na úrovni přímých instrukcí databáze. Toto není nutné, ale za některých okolností to může být výhodné. Program TESTEDITOR je vytvořen pro počítač ZX-Spectrum a jeho přenos na jiný počítač se nepředpokládá, protože databáze v něm vytvořené jsou funkčně nezávislé na typu cílového počítače!

Základní principy tvorby vlastního testu

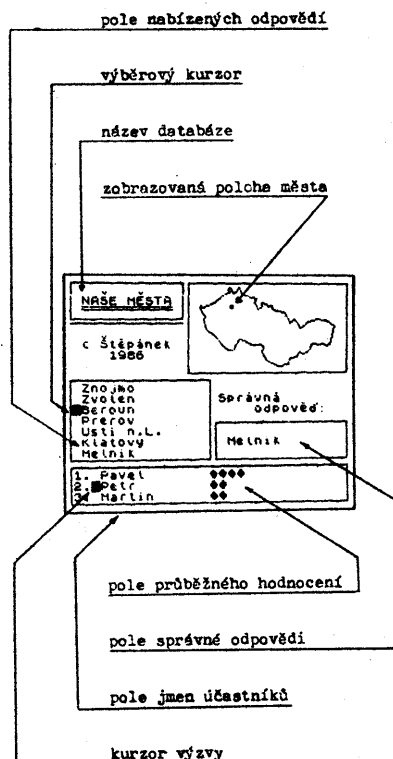
Každý test je tvořen jedním statickým obrázkem a sadou obrázků dynamických, které se postupně kreslí přes obrázek statický. Dynamický obrázek přitom vždy odpovídá jedné otázce testu. Při přechodu od aktuální otázky na následující se poslední dynamický obrázek smaže (při zachování statického) a vykreslí se obrázek nové otázky přes statický podklad.

Součástí každé otázky testu je vlastní otázka (slovní nebo grafická, jako např. zobrazení polohy města), několik nabízených odpovědí pro výběr a správná odpověď. Vlastní otázka je vždy zachycena v dynamickém obrázku (tedy napsána nebo nakreslena).

Způsob nabízení odpovědí je dvojitý. Při pevných odpovědích jsou tyto odpovědi součástí příslušného obrázku a jsou do něj přímo vepsány. Každá otázka zde má vždy stejné nabízené odpovědi. Při náhodných odpovědích nejsou odpovědi součástí obrázku a teprve v režimu zkoušení se generují a doplňují do dynamického obrázku. V tomto případě má stejná otázka při několika výskytech jiné nabízené odpovědi (jsou generovány náhodně z množiny všech možných odpovědí v bázi dat). Prostor nabízených odpovědí musí být na stínítku přesně vymezen, protože před texty jednotlivých odpovědí se pohybuje výběrový kurzor a v případě režimu náhodných odpovědí se prostor také musí mazat. Oblast je určena levým horním rohem, počtem nabízených odpovědí a maximální délkou odpovědí ve znacích.

Správná odpověď otázky testu zde figuruje jako název dynamického obrázku. Prostor pro vypsání správné odpovědi musí být opět přesně určen, a to levým rohem (odpověď je jen jedna a maximální délka je známa).

Statický obrázek tvoří neměnný podklad všech otázek. Musí respektovat oblasti pro nabízené odpovědi a pro zobrazení správné odpovědi. Navíc



Terminologie grafického uspořádání testů

musí mít vymezenou oblast pro zobrazení průběžného hodnocení účastníků. V této oblasti se v režimu zkoušení vypisují jména zkoušených a jejich aktuální hodnocení. Oblast je stanovena udáním levého horního rohu a počtem účastníků. Její délka je vždy 26 znaků. Na první pozici se pohybuje kurzor výzvy, na druhé pozici začíná jméno a na třinácté pozici je začátek oblasti hodnocení.

Statický obrázek i obrázky dynamické mohou využívat grafiky, textů a barev. Barvy použité v dynamickém obrázku by měly respektovat barvy statického obrázku v místech, do nichž se dynamický obrázek zobrazí. Nabízené odpovědi a správná odpověď se vypisují v naposled určených barvách v rámci dynamického obrázku. Jména

účastníků a jejich hodnocení naproti tomu zachovávají podkladovou barvu oblasti ve statickém obrázku.

Další parametry databáze jsou: doba čekání na odpověď (musí zahrnovat nejen dovolený čas na přemýšlení, ale také dobu potřebnou k manipulaci s výběrovým kurzorem) a mez špatného hodnocení. Tato mez je úspěšnost odpovědi účastníka, která již nevyhovuje požadavkům (slovní hodnocení ŠPATNĚ). Mez se určuje v procentech. Nejvyšší hodnocení SKVĚLE je pak pouze pro 100% úspěšnost. Interval mezi SKVĚLE a ŠPATNĚ je rozdělen na dva stejné podintervaly, které jsou značeny VÝBORNĚ a DOBŘE.

Hlavní menu programu TESTEDITOR

Po nahrání a spuštění programu se dostáváme do hlavního menu. Jsou zde příkazy pro nahrání databáze do počítače („l“-load), uchování databáze vytvořené editorem („s“-save) a ověření nahrávky databáze („v“-verify). Jako vnější paměť pro uvedené tři příkazy slouží buď magnetofon nebo ZX Microdrive.

Další příkaz „n“ umožňuje vybudovat novou databázi. Při jeho užití jsme dotázáni na barvu okraje, papíru a inkoustu a poté vstupujeme do vlastního grafického editoru, který je popsán v samostatné kapitole.

Příkazem pro doplnění databáze „d“ (doplň) můžeme k bázi dat nahrané pomocí load „l“ přidávat další obrázky, a to buď na konec nahrané databáze, nebo z nahrané databáze zachovat jen statický obrázek (který je možno ještě doplnit) a přidávat nové dynamické obrázky za něj. Druhá varianta je výhodná pro využití stejných základů statických obrázků pro více databází (např. obrysy Evropy pro databáze měst, moří a pohoří). Způsob doplnění datové báze se určuje v menu, které se objeví po užití příkazu „d“ hlavního menu.

Dále máme k dispozici příkaz k prohlížení hotových obrázků („p“-prohlížení). Při jeho užití se nás program zeptá, jestli si přejeme zobrazit podkladový statický obrázek nebo jen zobrazovat obrázky dynamické. Požadované obrázky se zadávají pořadovým číslem, přičemž číslo jedna odpovídá statickému obrázku. Maximální dovolené číslo obrázku je 80, protože čísla 81 až 100 jsou rezervována pro segmenty (1 až 20).

Příkazem volání monitoru „m“ (monitor) vyvoláme zvláštní program, který umožní pracovat s databází na úrovni jejich kódů. Monitor bude popsán také ve vlastní kapitole.

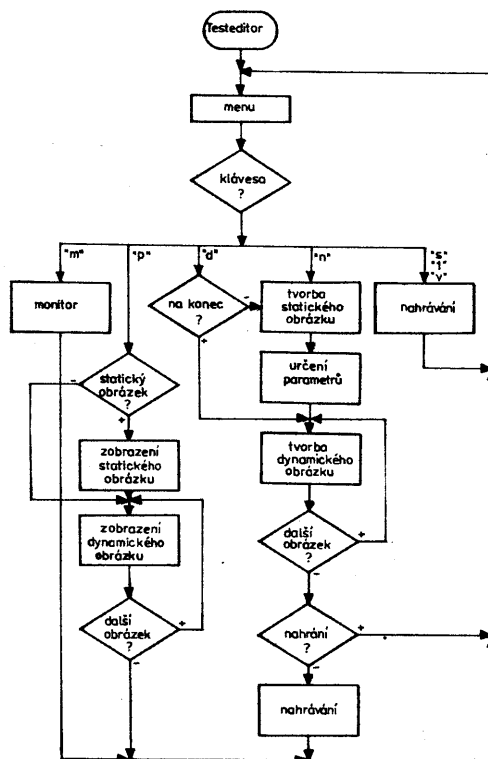
Pod nabídkou příkazů hlavního menu se zobrazují základní informace o současné databázi. Je to počet obrázků, délka a volný prostor. Zápis „30+1/2800“ znamená, že databáze má 30 dynamických obrázků a jeden statický, zabírá 2800 jednotek paměťového prostoru a zbývá 1200 jednotek volných pro další rozšiřování.

Grafický editor

Grafický editor umožňuje uživateli kreslit obrázky a automaticky vytvářet bázi dat. Do grafického editoru se dostáváme po užití příkazu „n“ a „d“ z hlavního menu. Seznam příkazů editoru a způsob jejich volání obdržíme po stisku klávesy „n“ (nápověda) v režimu editoru. Editor může pracovat v módu grafickém (jemná grafika 176 x 256 bodů) nebo v módu textovém (22 řádků po 32 znacích). Při vstupu do editoru jsme v režimu grafickém a do textového se můžeme dostat stiskem „t“ (text). V textovém módu můžeme užívat kurzory a zapisovat znaky. Zpět do grafického způsobu práce se dostaneme užitím klávesy ENTER. V grafickém režimu můžeme pohybovat grafickým kurzorem buď pomalu kurzorovými klávesami nebo rychle klávesami 5, 6, 7, 8 (tj. kurzory bez šiftu). Polohu grafického kurzoru můžeme také zadat hodnotově příkazem „h“ (hodnoty), který se dotáže na souřadnice x a y.

Bod vynese klávesou „o“ a kurzor spojíme s naposled vyneseným bodem úsečkou stiskem klávesy „ENTER“. Máme zde dále možnost nastavování barev okraje „o“, papíru „p“ a inkoustu „i“. Obrazovka se máže příkazem „v“ (výmaz), který zároveň nastaví naposled stanovené barvy pro celou plochu stínička. Jinak se totiž naposledy stanovené barvy uplatňují jen v následujících vyneseních bodu a znaků. Další možností je zápis příkazu zvukového signálu „z“ (zvuk) a blikání doposud nakresleného obrázku „b“. Pro lepší posouzení polohy bodů vzhledem k textovým čtverečkům slouží vynesení mřížky „m“. Pokud se dopustíme při kreslení chyby, můžeme poslední příkaz (nebo sérii příkazů) vymazat klávesou „SPACE“ (při mazání se vypíše název rušeného příkazu a jeho parametry).

Editor umožňuje definovat až 20 tzv. segmentů obrázku (značení číslem 1 až 20). Segmenty jsou vhodné pro opakování se částí dynamických obrázků. Pokud například jako segment nedefinujeme trojúhelník, můžeme v dynamických obrázcích testu dopravních značek volat tento segment pokřídě, když se má kreslit trojúhelníková značka. Nemusíme tedy trojúhelník stále znovu kreslit. Pro programátory je zřejmé chápání segmentu jako podprogramu. Segmenty mohou výrazně šetřit



Grafické schéma programu

paměť. Segment můžeme definovat tak, že na dotaz „Název obr. č.“ odpovíme znakem „“. Jsme dotázáni na číslo segmentu a poté převedeni do grafického editoru. Definici ukončíme stejně, jako by šlo o dynamický obrázek, tedy „k“. Segmenty můžeme definovat kdykoli v průběhu tvorby datové báze. V definici segmentů nesmíme ovšem užívat volání jiného segmentu! Nadefinované segmenty se v grafickém editoru volají příkazem „s“ (segment) a pak číslem segmentu. Abychom segmentem mohli nadefinovat obrazec, který bychom mohli při jeho volání umisťovat v různých polohách stínička, disponuje editor příkazem „r“ pro relativní spoj dvou bodů. Tak třeba v segmentu nadefinujeme čtverec užitím čtyř příkazů „r“ (bez vynesení bodu na začátku). Když potom v rámci některého obrázku vynese bod a zavoláme tento segment, vykreslí se nám čtverec v poloze určené vyneseným bodem. Je zde ovšem třeba jisté opatrnosti, abychom napří-

klad nevynesli bod do rohu stínička a nechťeli relativní obrazec vykreslit mimo hranice obrazovky. Pro tyto a některé další záludnosti příkazu „r“ doporučuji jeho užívání jen zkušeným uživatelům programu TESTEDITOR. Příkaz relativního spoje „r“ má oproti absolutnímu spoji „ENTER“ ještě jisté omezení, protože nejdelší úsečka smí být dlouhá nejvýše 127 bodů (delší úsečky vytvoříme nastavením). Režim grafického editoru opustíme stiskem „k“.

Vnitřní stavba databáze

Pro běžný návrh databází testů vystačíme s dosud poskytnutými informacemi. Pro některé rafinovanější vlastnosti bází nebo složitější opravy a úpravy však může být vhodné vědět něco o její struktuře. Tyto znalosti jsou také nezbytné pro práci s monitorem. Databáze je tvořena přístupovým vektorem p(), hlavním informačním vektorem b\$() a třinácti pomocnými proměnnými. Obrázky jsou v databázi rozloženy do sérií operací. Jsou tedy vlastně určeny jakýmsi programem ve zvláštním jazyce, jehož příkazy jsou uvedeny dále. Program PROFESOR (verze 1.0) a program TESTEDITOR (verze 1.0) rozumí příkazům z tab. 1.

Operace s parametry jsou uloženy ve vektoru b\$(). Začátky jednotlivých obrázků v b\$() jsou označeny odkazy uloženými ve vektoru p(), jehož index představuje číslo obrázku (tj. když p(3)=2086, tak obr. 3 začíná v b\$(2086) od buňky s indexem 2086). Délka sekvence příslušející obrázku je dána obsahem buňky pod ukazatelem začátku a buňky v b\$(256) následující (256 * první buňka + druhá buňka).

Použití MONITORU

Monitor slouží pro přímé prohlížení a modifikaci báze dat. Odkazy přístupového vektoru vypíšeme příkazem „p“, číselný výpis databáze získáme příkazem „c“ (číselný) a slovní výpis operací s parametry příkazem „o“ (operace). Číselný obsah můžeme změnit voláním příkazu změny „z“. V tomto režimu můžeme psát čísla, která přepíší původní obsah nebo stiskneme jen „ENTER“ a obsah buňky zůstane beze změny. Ukončení režimu se děje zápisem znaku „k“ (konec). Abychom poznali, které části obrázku přísluší určité operace nebo naopak, můžeme použít režimu trasování kresby (příkaz „t“). V režimu trasování krokujeme kresbu klávesou „k“ nebo klávesou „t“ můžeme provádět krokované zároveň s výpisem právě prováděné operace. Trasování můžeme přerušit podržením klávesy „SPACE“. Monitor opouštíme stiskem „k“ (konec) a tím se navrátíme do hlavního menu programu.

Tab. 1. Příkazy

Kód	Parametry	Význam
0	délka, název	— načtení názvu dynamického obrázku
1	x, y	— vynesení bodu v místě se souřadnicemi x, y
2	x, y	— spojení naposled vyneseného bodu s x, y
3	řádek, sloupec	— nastavení textového kurzoru
4	délka, znak ...	— vypsání řetězce znaků
5	barva	— nastavení barvy inkoustu *
6	barva	— nastavení barvy papíru *
7		— výmaz stínička a nastavení barev
8		— krátký zvukový signál
9	dx, dy	— spoj od naposled vyneseného bodu do x+dx, y+dy
10	číslo segmentu	— vykreslení segmentu obrázku
11	barva	— nastavení barvy vnějšího okraje *
12	počet	— blikání dosud nakreslené části
13	číslo	— načtení čísla pozice správné odpovědi

* Kódy barev jsou následující: 0 — černá, 1 — tmavě modrá, 2 — červená, 3 — fialová, 4 — zelená, 5 — světle modrá, 6 — žlutá, 7 — bílá (kódování podle mateřského počítače Sinclair ZX Spectrum).

Výpis programu „Testeditor“

```

1 GO TO 7000: REM Studeny sta
2 GO TO 7001: REM Morly start
10 PRINT 0; PAPER 0; INK 7; AT
0; 0; INVERSE 1; "UOLNYCH: "; 400
0-60; "REZIM: "; 1; 0; 0;
11 PRINT 0; AT 1,25; " "; PAPER
R 10; INK 7+(10/4); 1; 10; PAPER
10; INK 7+(10/4); 1; 10; PAPER
12 PRINT 0; AT 1,0; "x="; x; y
=""; y; "x="; x; y; "y="; y; "x="; x; y
13 LET k=1
14 IF INKEY$="" THEN LET a=35
15 IF INKEY$="" AND a>0 THEN
LET a=a-1: GO TO 15
16 IF INKEY$="" THEN GO TO 16
18 LET k$=INKEY$
20 IF k$="TEXTOUY" THEN GO TO
50
22 IF k$=CHR$ 8 THEN GO SUB 10
0: GO TO 12: REM vlevo
24 IF k$=CHR$ 9 THEN GO SUB 11
0: GO TO 12: REM vpravo
26 IF k$=CHR$ 10 THEN GO SUB 1
20: GO TO 12: REM dolu
27 IF k$=CHR$ 11 THEN GO SUB 1
30: GO TO 12: REM nahoru
28 IF k$="0" THEN LET k=16: GO
SUB 100: LET k=1: GO TO 12
29 IF k$="1" THEN LET k=16: GO
SUB 110: LET k=1: GO TO 12
30 IF k$="2" THEN LET k=16: GO
SUB 120: LET k=1: GO TO 12
31 IF k$="3" THEN LET k=16: GO
SUB 130: LET k=1: GO TO 12
32 IF k$=CHR$ 13 THEN GO SUB 8
00
33 IF k$=" " THEN GO SUB 650
34 IF k$="." THEN GO SUB 740
35 IF k$="," THEN GO SUB 750
36 IF k$=";" THEN GO SUB 710
37 IF k$=":" THEN GO SUB 730
38 IF k$="@" THEN GO SUB 900
39 IF k$="h" THEN GO SUB 790
40 IF k$="z" THEN GO SUB 550
41 IF k$="k" THEN GO SUB 660:
RETURN
42 IF k$="0" THEN GO SUB 600
43 IF k$="n" THEN GO SUB 6500
44 IF k$="i" THEN GO SUB 620
45 IF k$="a" THEN GO SUB 6500
46 IF k$="p" THEN GO SUB 640
47 IF k$="v" THEN GO SUB 660
48 IF k$="b" THEN GO SUB 670
49 GO TO 10
50 IF k$=CHR$ 8 THEN LET p=1:
GO SUB 200: GO TO 12: REM vlevo
52 IF k$=CHR$ 9 THEN LET p=1:
GO SUB 220: GO TO 12: REM vpravo
54 IF k$=CHR$ 10 THEN LET p=1:
GO SUB 240: GO TO 12: REM dolu
56 IF k$=CHR$ 11 THEN LET p=1:
GO SUB 260: GO TO 12: REM nahor
U
57 IF k$=CHR$ 13 THEN LET p=1:
LET k$="GRAFICKY": PRINT AT y,
x; OVER 1; PAPER 8; INK 8; " ":
GO TO 10
58 PRINT AT y, x; k;
62 IF p=1 THEN GO SUB 80
64 LET x=x+1
66 IF x>31 THEN LET x=0
67 PRINT AT y, x; OVER 1; FLA
SH 1; PAPER 8; INK 8; " "
68 LET b$(b0)=k$
69 LET b0=b0+1
70 LET b$(p1)=CHR$ (CODE b$(p1
)+1)
GO TO 12
80 LET b$(b0)=CHR$ 3
81 LET b$(b0+1)=CHR$ y
82 LET b$(b0+2)=CHR$ x
83 LET b$(b0+3)=CHR$ 4
84 LET p1=b0+4
85 LET b$(p1)=CHR$ 0
86 LET b0=b0+5
88 LET p0=0
89 RETURN
90 REM vlevo
100 IF k0=0 THEN PLOT OVER 1; P
APER 8; INK 8; x,y
101 LET x=x-k
102 IF x<0 THEN LET x=0
103 LET k0=0
104 PLOT OVER 1; PAPER 8; INK 8
; x,y
105 RETURN
106 REM vpravo
110 IF k0=0 THEN PLOT OVER 1; P
APER 8; INK 8; x,y
111 LET x=x+k
112 IF x>255 THEN LET x=255
114 LET k0=0
116 PLOT OVER 1; PAPER 8; INK 8
; x,y
117 RETURN
118 REM dolu
120 IF k0=0 THEN PLOT OVER 1; P
APER 8; INK 8; x,y
121 LET y=y-k
122 IF y<0 THEN LET y=0
124 LET k0=0
126 PLOT OVER 1; PAPER 8; INK 8
; x,y
128 RETURN

```

```

129 REM nahoru
130 IF k0=0 THEN PLOT OVER 1; P
APER 8; INK 8; x,y
131 LET y=y+k
132 IF y>175 THEN LET y=175
134 LET k0=0
136 PLOT OVER 1; PAPER 8; INK 8
; x,y
138 RETURN

```

```

199 REM Text vlevo
200 PRINT AT y, x; OVER 1; PAP
ER 8; INK 8; " "
201 LET x=x-1
202 IF x<0 THEN LET x=0
203 PRINT AT y, x; OVER 1; FLA
SH 1; PAPER 8; INK 8; " "
204 RETURN

```

```

219 REM Text vpravo
220 PRINT AT y, x; OVER 1; PAP
ER 8; INK 8; " "
221 LET x=x+1
222 IF x>31 THEN LET x=31
223 PRINT AT y, x; OVER 1; FLA
SH 1; PAPER 8; INK 8; " "
224 RETURN
229 REM Text dolu
230 PRINT AT y, x; OVER 1; PAP
ER 8; INK 8; " "
231 LET y=y+1
232 IF y>31 THEN LET y=21
233 PRINT AT y, x; OVER 1; FLA
SH 1; PAPER 8; INK 8; " "
234 RETURN
259 REM Text nahoru
260 PRINT AT y, x; OVER 1; PAP
ER 8; INK 8; " "
261 LET y=y-1
262 IF y<0 THEN LET y=0
263 PRINT AT y, x; OVER 1; FLA
SH 1; PAPER 8; INK 8; " "
264 RETURN

```

```

579 REM zvukovy signal
580 LET b$(b0)=CHR$ 8
581 LET b0=b0+1
582 BEEP 0.1,30
584 RETURN

```

```

599 REM Border
600 INPUT "Barva okraje="; j
601 BORDER j
602 LET b$(b0)=CHR$ 11
603 LET b$(b0+1)=CHR$ j
604 LET b0=b0+2
605 RETURN
610 REM Inkoust
611 INPUT "Barva inkoustu="; j
612 INK j
613 LET b$(b0)=CHR$ 5
614 LET b$(b0+1)=CHR$ j
615 LET b0=b0+2
616 RETURN
630 REM Papir
640 INPUT "Barva papiru="; j
641 PAPER j
642 LET b$(b0)=CHR$ 6
643 LET b$(b0+1)=CHR$ j
644 LET b0=b0+2
645 RETURN
650 REM Uymaz
660 CLS
661 LET b$(b0)=CHR$ 7
662 LET b0=b0+1
663 RETURN
669 REM Blikani
670 INPUT "Kolikrat="; j
671 LET b$(b0)=CHR$ 12
672 LET b$(b0+1)=CHR$ j
673 LET b0=b0+2
674 RETURN

```

```

659 REM Uymaz
660 CLS
661 LET b$(b0)=CHR$ 7
662 LET b0=b0+1
663 RETURN
669 REM Blikani
670 INPUT "Kolikrat="; j
671 LET b$(b0)=CHR$ 12
672 LET b$(b0+1)=CHR$ j
673 LET b0=b0+2
674 RETURN

```

```

699 REM Hodnota
700 OVER 1: PLOT x,y
701 INPUT "x="; x

```

```

702 INPUT "y="; y
703 PLOT x,y: OVER 0
706 RETURN

```

```

709 REM Plot
710 PLOT OVER 0; x,y
711 LET k0=1
712 LET x=x
713 LET y=y
714 LET b$(b0)=CHR$ 1
715 LET b$(b0+1)=CHR$ x
716 LET b$(b0+2)=CHR$ y
717 LET b0=b0+3
724 RETURN

```

```

729 REM Text v
730 LET k$="TEXTOUY"
731 PRINT AT y, x; OVER 1; FLA
SH 1; PAPER 8; INK 8; " "
732 RETURN

```

```

739 REM Rel draw
740 IF ABS (x-x1)>127 OR ABS (y
-y1)>127 THEN PRINT 0; AT 1,0; 0
741 RETURN
742 PLOT xs,ys: DRAW x-xs,y-ys
743 LET b$(b0)=CHR$ 9
744 LET b$(b0+1)=CHR$ (x-xs+126
)
745 LET b$(b0+2)=CHR$ (y-ys+126
)
746 GO TO 808
749 REM Segment
750 LET u5=p0: LET u4=u: LET u3
=b0
752 INPUT "Cisto segmentu (1-20
)"; i
753 LET b$(b0)=CHR$ 10: LET b$(
b0+1)=CHR$ 0: LET o=0+80
754 GO SUB 3000
755 LET p0=u5: LET u=u4: LET b0
=u3
756 LET b0=b0+2
759 RETURN

```

```

799 REM Draw
800 PLOT OVER 0; xs,ys
801 DRAW OVER 0; x-xs,y-ys
802 LET b$(b0)=CHR$ 2
803 LET b$(b0+1)=CHR$ x
804 LET b$(b0+2)=CHR$ y
805 LET b0=b0+3
806 LET xs=x
807 LET ys=y
808 LET u=0
814 RETURN

```

```

849 REM Pseudoplot
850 PLOT OVER 0; x,y
852 LET xs=x
854 LET ys=y
856 LET k0=1
858 RETURN

```

```

899 REM Mazani
900 CLS
901 LET i=p(p0-1)+2
902 IF (9)>80 THEN LET i=p(f9)+
2
904 LET i0=1
905 LET j=CODE b$(i)

```

```

910 IF j=7 OR j=8 OR j=32 THEN
LET i=i+1
915 IF j=1 OR j=2 OR j=3 OR j=9
THEN LET i=i+3
920 IF j=5 OR j=6 OR j=10 OR j=
11 OR j=12 OR j=13 THEN LET i=i+
2
921 IF j=0 OR j=4 THEN LET i=i+
2+CODE b$(i+1)
922 IF i<0 THEN GO TO 904
925 LET b0=i0: LET u=i0: GO SUB
6725: BEEP 0.1,30
927 IF k=0 THEN GO TO 930
928 LET u=u+1: LET k=k-1: PRINT
CODE b$(u);
929 GO TO 927
930 IF b0=p(p0-1)+2 AND p0>2 TH
EN LET p0=p0-1: LET b0=p(p0-1)+2
931 INPUT "Dalsi mazani? (a/n)
"; k$
932 LET f7=0: IF k$="a" THEN LE
T f7=1: GO SUB 935: GO TO 901

```

```

935 REM Aktualizace obrazku
936 LET j=b0-p(p0-1)-2
937 IF (9)>80 THEN LET j=b0-p(f
9)-2: LET b$(p(f9))=CHR$ (INT (j
-255)): LET b$(p(f9)+1)=CHR$ (j
-255+INT (j/255)): GO TO 943
941 LET b$(p(p0-1))=CHR$ (INT (
j/255))
942 LET b$(p(p0-1)+1)=CHR$ (INT
(j-255+INT (j/255)))
943 IF f7=1 THEN RETURN
944 LET o=1
945 GO SUB 3000
946 LET o=p0-1: IF (9)>80 THEN
LET o=f9
947 IF o<1 THEN GO SUB 3000
948 PLOT OVER 1; PAPER 8; INK 8
; x,y: LET k0=0
949 LET k$=""
950 RETURN
950 IF k0=0 THEN PLOT OVER 1; P
APER 8; INK 8; x,y
952 RETURN
1000 CLS
1001 DIM x(13)
1002 DIM p(100)
1003 DIM b$(4000)
1004 DIM x(704)
1005 LET x=0: LET xs=x
1006 LET y=0: LET ys=y
1007 LET x1=0: LET y1=y
1008 LET k0=0
1009 LET k$="GRAFICKY"
1010 LET p0=2
1011 LET b0=3
1012 LET k=1
1013 LET f6=0
1014 LET f8=0
1015 LET f1=0
1016 LET f9=0
1017 LET p=1
1018 LET f0=0
1019 LET f7=0
1020 LET i6=7
1021 LET i9=1
1022 RETURN

```



```

1024 REM Nova database
1025 LET P(1)=1
1026 PRINT "Vytvoreni statickeho
obrazku"
1030 INPUT "Barva okraje=";J
1032 LET b$(0)=CHR$(J)
1034 LET b$(b0+1)=CHR$(J)
1036 INPUT "Barva papiru=";J
1038 LET b$(b0+2)=CHR$(J)
1040 LET b$(b0+3)=CHR$(J)
1042 INPUT "Barva inkoustu=";J
1044 LET b$(b0+4)=CHR$(J)
1046 LET b$(b0+5)=CHR$(J)
1048 BEEP 0.1,30
1049 BORDER CODE b$(b0+1)
1050 PAPER CODE b$(b0+3)
1051 INK CODE b$(b0+5)
1052 LET b$(b0+6)=CHR$(J)
1055 CLS
1056 LET b0=b0+7
1057 PLOT 0,0
1058 GO SUB 10
1060 BEEP 0.1,30
1062 LET i=b0-3
1064 LET b$(1)=CHR$(INT (1/256)
)
1066 LET b$(2)=CHR$(1-256*INT (
1/256))
1068 INPUT "Nevne nebo nahodne o
dpovedi?";k$
1070 LET q=0
1072 IF k$="p" THEN LET q=1
1073 GO SUB 5500
1074 PRINT #0;AT 0,0;"Levy horni
roh nabidky odpovedi:"
1075 PAUSE 200
1076 INPUT "radek=";r1
1078 INPUT "sloupec=";s1
1080 INPUT "Max.delka odpovedi="
;
1082 INPUT "Pocet nabizenych odp
ovedi=";n
1084 FOR i=r TO r+n-1
1086 PRINT AT 1,s-1;i-r+1;
1088 FOR j=1 TO d
1090 PRINT "0";
1092 NEXT j
1094 NEXT i
1096 PRINT #0;AT 0,0;"Levy roh s
pravne odpovedi:"
1097 PAUSE 200
1098 INPUT "radek=";r1
1100 INPUT "sloupec=";s1
1102 PRINT AT r1,s1;
1104 FOR i=1 TO d
1106 PRINT "5";
1108 NEXT i
1110 PRINT #0;AT 0,0;"Levy horni
roh hodnoceni:"
1111 PAUSE 200
1112 INPUT "radek=";r0
1114 INPUT "sloupec=";s0
1116 INPUT "Pocet ucastniku (max
5)=";J
1117 LET n=J
1118 FOR i=1 TO J
1120 PRINT AT (0+i-1,s0);"JMENO.
.....HODNOCENI...."
1122 NEXT i
1130 INPUT "Doba cekani na odpov
ed [s]=";t
1132 PRINT #0;AT 0,0;"Mez spatne
hodnoceni=";
1133 PAUSE 200
1134 INPUT "Mez (nahodne)";(INT
(100/n))
1136 PRINT #0;AT 0,0;
1137 PAUSE 0
1138 CLS: LET o=1: GO SUB 3000:
BEEP 0.1,30
1149 REM Dynamicke obrázky
1150 LET p(p0)=b0
1152 LET p0=p0+1
1154 LET b0=b0+2
1156 LET x=0: LET y=0: LET x1=0:
LET y1=0: LET xs=0: LET ys=0
1158 INPUT "Název obr.":(p0-1);"
";k$: PRINT AT r1,s1;k$
1160 LET i=0: IF k$="1" THEN GO
SUB 9500: GO TO 1173
1162 LET b$(b0)=CHR$(0)
1164 LET b$(b0+1)=CHR$(LEN k$)
1166 LET b0=b0+1
1168 FOR i=1 TO LEN k$
1170 LET b$(b0+i)=k$(i TO i)
1172 NEXT i
1174 LET b0=b0+LEN k$+1
1176 PLOT 0,0
1178 GO SUB 10
1180 IF q=0 OR (q>=80 THEN GO TO
1176
1182 INPUT "Cislo spravne odpove
di=";k
1184 LET b$(b0)=CHR$(13)
1186 LET b$(b0+1)=CHR$(k)
1188 LET b0=b0+2
1190 IF (q>=80 THEN GO SUB 9500:
GO TO 1182
1192 LET b0=p(p0-1)-2
1194 LET b$(p(p0-1))=CHR$(INT (
J/256))
1196 LET b$(p(p0-1)+1)=CHR$(INT
(J-256*INT (J/256)))
1198 INPUT "Dalsi dynamicky obr?
(a/n)";k$
1199 IF k$="a" THEN GO SUB 3000:
GO TO 1150
1200 INPUT "Nahrani database? (a
/n)";k$
1202 IF k$="a" THEN GO SUB 6000
1204 RETURN
3000 IF o=0 THEN BEEP 1,20: RETU
RN
3001 LET u=p(o): IF u=0 THEN BEE
P 1,20: RETURN
3002 LET u0=u+256*CODE b$(u)+COD
E b$(u+1)+1
3004 LET u=u+2
3005 PRINT #0;AT 1,0;">>> Klave
s: <<<< preruste <<<<: BEEP 0.
02,30: PAUSE 18*50+1
3006 IF INKEY$="" THEN LET u=u0
+1
3007 IF (o=1 THEN GO SUB 8950
3010 IF u>u0 AND o<1 THEN LET i
=1
3011 IF u>u0 THEN LET i=0: LET i
=0: RETURN
3012 LET k=CODE b$(u)
3013 LET u=u+1
3014 IF k=2 THEN GO SUB 3600: GO
TO 3006
3016 IF k=1 THEN GO SUB 3610: GO
TO 3006
3018 IF k=0 THEN GO SUB 3750: GO
TO 3006
3020 IF k=3 THEN GO SUB 3620: GO
TO 3006
3022 IF k=4 THEN GO SUB 3630: GO
TO 3006
3023 IF k=9 THEN GO SUB 3680: GO
TO 3006
3024 IF k=5 THEN GO SUB 3640: GO
TO 3006
3025 IF k=10 THEN GO SUB 3690: GO
TO 3006
3026 IF k=6 THEN GO SUB 3650: GO
TO 3006
3028 IF k=7 THEN GO SUB 3660: GO
TO 3006
3030 IF k=8 THEN GO SUB 3670: GO
TO 3006
3032 IF k=11 THEN GO SUB 3710: GO
TO 3006
3034 IF k=12 THEN GO SUB 3720: GO
TO 3006
3035 IF k=32 THEN GO TO 3006
3036 IF k=13 THEN LET so=CODE b$
(u): LET u=u+1: GO TO 3006
3037 CLS
3038 PRINT "CHYBNY KOD: "
3040 PRINT "Obr. ";o;" v b$ od "
;P(0)
3042 PRINT "Pozice v b$ ";u-1
3044 STOP
3599 REM Kresleni
3600 DRAW CODE b$(u)-xs, CODE b$(
u+1)-ys
3602 LET xs=CODE b$(u)
3604 LET ys=CODE b$(u+1)
3606 LET u=u+2
3608 RETURN
3610 PLOT CODE b$(u), CODE b$(u+1)
3612 LET xs=CODE b$(u)
3614 LET ys=CODE b$(u+1)
3616 LET u=u+2
3618 RETURN
3620 PRINT AT CODE b$(u), CODE b$
(u+1);
3622 LET u=u+2
3624 RETURN
3630 FOR i=u+1 TO u+CODE b$(u)
3632 PRINT b$(i);
3634 NEXT i
3636 LET u=u+CODE b$(u)+1
3638 RETURN
3640 INK CODE b$(u)
3642 LET u=u+1
3644 RETURN
3650 PAPER CODE b$(u)
3652 LET u=u+1
3654 RETURN
3660 CLS
3662 RETURN
3670 BEEP 0.1,30
3672 RETURN
3680 DRAW CODE b$(u)-128, CODE b$
(u+1)-128
3682 LET xs=xs+CODE b$(u)-128
3684 LET ys=ys+CODE b$(u+1)-128
3686 LET u=u+2
3688 RETURN
3689 REM Segment
3690 LET u5=0: LET u4=u0: LET u3
=u0
3692 LET o=CODE b$(u)+80
3694 GO SUB 3000
3696 LET o=u5: LET u0=u4: LET u=
u3+1
3698 RETURN
3700 LET i=0: LET n0=0: IF i=1
FOR 1 THEN RETURN
3701 LET u5=0: LET u3=u0: LET u4=
u0
3702 LET i=1
3703 OVER 1
3704 GO SUB 3000
3705 OVER 0
3706 LET o=u5: LET u=u3: LET u0=
u4
3707 LET i=0
3708 RETURN
3710 BORDER CODE b$(u)
3712 LET u=u+1
3714 RETURN
3720 OVER 1
3722 IF i=0 THEN LET i=1: LET
n0=CODE b$(u)
3724 IF n0=0 THEN LET i=0: OVER
0
3725 LET u=u+1: RETURN
3726 LET n0=n0-1
3728 LET u=p(o)
3729 GO SUB 3000
3730 LET u=p(o)+CODE b$(p(o))
3732 RETURN
3734 LET n$=""
3736 FOR i=1 TO CODE b$(u)
3738 LET n$=n$+b$(u+1)
3740 NEXT i
3742 PRINT AT r1,s1;n$
3744 LET u=u+CODE b$(u)+1
3746 RETURN
3800 LET i=1: LET f2=1: LET f3=
0: LET j=1
3801 IF (f0=80 THEN CLS: LET o=
1: GO SUB 3000: GO TO 3804
3802 LET o=p0-1: LET u=p(o)+3
3803 GO SUB 3700
3804 LET f1=0: LET f2=0
3806 RETURN

```

```

3999 REM Datodovani
4000 LET k=CODE b$(u)
4002 IF k=0 THEN LET k$="Ovn.naz
sv": LET k=CODE b$(u+1)+1: RETUR
N
4004 IF k=1 THEN LET k$="Bod": L
ET k=2: RETURN
4006 IF k=2 THEN LET k$="Usecka"
: LET k=3: RETURN
4008 IF k=3 THEN LET k$="Kurzor"
: LET k=4: RETURN
4010 IF k=4 THEN LET k$="Text":
LET k=CODE b$(u+1)+1: RETURN
4012 IF k=5 THEN LET k$="Inkoust"
: LET k=6: RETURN
4014 IF k=6 THEN LET k$="Papir":
LET k=7: RETURN
4016 IF k=7 THEN LET k$="Uvaz":
LET k=8: RETURN
4018 IF k=8 THEN LET k$="Zvuk.si
gnal": LET k=9: RETURN
4019 IF k=9 THEN LET k$="Nic":
LET k=0: RETURN
4020 IF k=10 THEN LET k$="Okraj"
: LET k=11: RETURN
4022 IF k=11 THEN LET k$="Blitan
i": LET k=12: RETURN
4024 IF k=12 THEN LET k$="Spravn
a odpoved": LET k=13: RETURN
4026 IF k=13 THEN LET k$="Rel.use
cka": LET k=14: RETURN
4028 IF k=14 THEN LET k$="Segmen
t": LET k=15: RETURN
4030 LET k$="Chyba!"
4032 LET k=0
4034 RETURN

```

```

5999 REM Ulozeni database
6000 CLS
6002 PRINT "Volte pametove zariz
eni:"
6003 PRINT "1-kazetovy mgi"
6004 PRINT "2-microdrive"
6006 INPUT k$: CLS
6008 LET x(1)=p0-1
6010 LET x(2)=n
6012 LET x(3)=m
6014 LET x(4)=r
6016 LET x(5)=s
6018 LET x(6)=0
6020 LET x(7)=0
6022 LET x(8)=t
6024 LET x(9)=d
6026 LET x(10)=v
6028 LET x(11)=q
6030 LET x(12)=r1
6032 LET x(13)=s1
6034 INPUT "Název database=";j$
6036 IF k$="1" THEN GO TO 6050
6038 SAVE "a";j$;"1" DATA p()
6040 SAVE "a";j$;"2" DATA b()
6042 SAVE "a";j$;"3" DATA x()
6044 RETURN
6046 SAVE "a";j$;"1" DATA p()
6048 SAVE "a";j$;"2" DATA b()
6050 SAVE "a";j$;"3" DATA x()
6052 LET k$=""
6054 RETURN

```

```

6099 REM Nahrani database
6100 CLS
6102 PRINT "Volte pametove zariz
eni:"
6103 PRINT "1-kazetovy mgi"
6104 PRINT "2-microdrive"
6106 INPUT k$: CLS
6108 INPUT "Název database=";j$
6110 IF k$="1" THEN GO TO 6120
6112 LOAD "a";j$;"1" DATA p()
6114 LOAD "a";j$;"2" DATA b()
6116 LOAD "a";j$;"3" DATA x()
6118 GO TO 6150
6120 LOAD j$;"1" DATA p()
6122 LOAD j$;"2" DATA b()
6124 LOAD j$;"3" DATA x()
6126 LET p0=p(p0-1)+256*CODE b$(
p(p0-1))+CODE b$(p(p0-1)+1)+2
6128 LET n=x(2)
6130 LET m=x(3)
6132 LET r=x(4)
6134 LET s=x(5)
6136 LET v=x(6)
6138 LET q=x(7)
6140 LET d=x(8)
6142 LET t=x(9)
6144 LET x(10)=v
6146 LET x(11)=q
6148 LET x(12)=r1
6150 LET x(13)=s1
6152 LET k$=""
6154 RETURN

```

```

6199 REM Overeni database
6200 CLS
6202 PRINT "Volte pametove zariz
eni:"
6203 PRINT "1-kazetovy mgi"
6204 PRINT "2-microdrive"
6206 INPUT k$: CLS
6208 INPUT "Název database=";j$
6210 IF k$="1" THEN GO TO 6220
6212 VERIFY "a";j$;"1" DATA p
()
6214 RETURN

```

```

6216 VERIFY "M";J$+"2" DATA b
6216 VERIFY "M";J$+"3" DATA x
6219 RETURN
6220 PRINT "Hledam: ";J$+"1"
6222 VERIFY J$+"1" DATA p(1)
6224 PRINT "Hledam: ";J$+"2"
6226 VERIFY J$+"2" DATA b$(1)
6228 PRINT "Hledam: ";J$+"3"
6230 VERIFY J$+"3" DATA x(1)
6231 LET k$=""
6232 RETURN

```

```

6499 REM Mrizka
6500 PRINT AT 0,0
6520 FOR i=0 TO 10
6522 PRINT PAPER 5; INK 0; OVER
1;

```

```

REM Dvouradkova sachovnice
s bright
6530 NEXT i
6531 BORDER 5
6532 RETURN

```

```

6599 REM Napoveda
6600 PAPER 1; INK 7; BORDER 1; C
LS
6602 PRINT "Seznam prikazu edito
ru..."
6604 PRINT "F1 - rychly graf
icky kurzor"
6606 PRINT "F2 - presny text
ovy a gra-
fy"
6609 PRINT "F3 - hodnotove zadani
kurzoru"
6610 PRINT "F4 - vyneseni bodu pod
kurzorem"
6612 PRINT "F5 - spoj kurzoru
s posl.bodem"
6614 PRINT "F6 - nastaveni barvy o
kraje"
6616 PRINT "F7 - nastaveni barvy p
apiru"
6618 PRINT "F8 - nastaveni barvy i
nkoustu"
6620 PRINT "F9 - vymaz obr. a nast
aveni barev"
6622 PRINT "F10 - textovy rezim (zp
et F1)"
6624 PRINT "F11 - zvukovy signal"
6626 PRINT "F12 - blikani dosud vyt
voreného obr."
6627 PRINT "F13 - volani segmentu o
brázku"
6628 PRINT "F14 - pomocna mrizka (z
pet)"
6630 PRINT "F15 - zruseni posle
dní operace"
6631 PRINT "F16 - napoveda (=toto me
nu)"
6632 PRINT "F17 - (relativni spoj p
ro segmenty)"
6633 PRINT "F18 - (pseudobod pro re
lativ. spoj)"
6634 PRINT "F19 - konec prace v edi
toru"
6635 PRINT #0; "Konec programu"
6636 PAUSE 0
6637 CLS
6638 GO SUB 935
6640 RETURN

```

```

6699 REM Doplneni
6700 CLS
6701 PRINT "Zpusob doplneni: "; P
RINT
6702 PRINT "1 - dynam obr na kone
c database"
6703 PRINT "2 - staticky obr se s
mazanim vseh dynamickych"
6705 INPUT k$: CLS
6706 LET o=1: GO SUB 3000
6707 IF k$="d" THEN GO TO 6710
6709 LET p0=2: LET b0=p(2): GO S
UB 1057: GO TO 6713
6712 GO SUB 1150
6713 LET k$=""
6714 RETURN

```

```

6999 REM Hl.menu
7000 BEEP 0.1,30: GO SUB 1000
7001 BORDER 1: PAPER 1; INK 7; C
LS
7002 PRINT " * * T E S T E D I
T O R * * "
7003 PRINT "=====
=====
c Martin Stepane
k 1985"
7005 PRINT " (verze 1.0)
"
7007 PRINT
7008 PRINT
7009 PRINT "1 - load database"
7010 PRINT "2 - save database"
7012 PRINT "3 - verify database"
7014 PRINT "4 - nova database"
7016 PRINT "5 - doplneni database"
7018 PRINT "6 - prohlizeni databa
ze"
7020 PRINT "7 - monitor"
7022 PRINT "8 - konec programu"
7024 PRINT AT 20,0;"Baze: "; INU

```

```

ERSE 1,p0-2,"+1/";b0;" (zbyva "
4000-b0;"")
7026 LET i8=0
7031 INPUT "Volte rezim: ";k$
7032 CLS: BEEP 0.1,30
7034 IF k$="s" THEN GO SUB 6000
7036 IF k$="l" THEN GO SUB 6100
7038 IF k$="v" THEN GO SUB 6200
7040 IF k$="m" THEN GO SUB 6500
7042 IF k$="p" THEN GO SUB 6800
7044 IF k$="n" THEN GO SUB 1000:
GO SUB 1025
7046 IF k$="d" THEN GO SUB 6700
7050 IF k$="k" THEN STOP
7052 GO TO 7001

```

```

8499 REM Monitor
8500 BORDER 1: PAPER 1; INK 7; C
LS
8502 PRINT "Monitor-vyberte rezi
m..."
8503 PRINT
8504 PRINT "1 - vypis pristupoveh
o vektoru p"
8506 PRINT "2 - ciselny vypis baz
e b$"
8508 PRINT "3 - vypis operaci baz
e b$"
8510 PRINT "4 - zmena obsahu b$"
8511 PRINT "5 - trasovani kresby
obr."
8514 PRINT "6 - konec prace s mon
itorem"
8515 INPUT k$: BEEP 0.1,30
8516 IF k$="c" THEN GO SUB 9000
8517 IF k$="o" THEN GO SUB 8700
8518 IF k$="p" THEN GO SUB 8600
8520 IF k$="z" THEN GO SUB 8750
8521 IF k$="t" THEN GO SUB 8900
8523 IF k$="k" THEN LET k$="": R
ETURN
8524 PRINT #0; AT 1,0;"Stisknete
klavesu!"
8525 GO TO 8500

```

```

8599 REM Pristup vektor
8600 CLS
8601 FOR i=1 TO 100
8602 PRINT i,p(i)
8604 NEXT i
8606 RETURN

```

```

8699 REM Vypis operaci
8700 INPUT "Cislo obr.(u segment
u +80) = ";o: CLS: IF o<1 OR o>10
0 THEN BEEP 1,20: RETURN
8701 IF p(o)=0 THEN BEEP 1,20: R
ETURN
8702 LET j=256+CODE b$(p(o))+COD
E b$(p(o)+1)+2
8703 PRINT "Vypis operaci obr. "
o;" (delka=";j;"")
8704 PRINT #0; AT 1,0;">>> Klaves
a 1-9 prerusuje <<<"
8705 FOR i=p(o)+2 TO p(o)+j-1
8706 LET u=i
8708 GO SUB 8726
8710 LET i=u
8720 IF INKEY$="" THEN LET i=32
8722 NEXT i
8724 RETURN

```

```

8725 REM Vypis operace
8726 GO SUB 4000
8727 IF i8=0 THEN PRINT u;"-";k
$;"
8728 IF i8=1 THEN PRINT #0; AT 1,
0,u;"-";k$;"
8731 IF i8=1 AND k>3 THEN PRINT
#0;CODE b$(u+1);"...": GO TO 6
738
8732 IF k=0 THEN GO TO 8738
8733 LET u=u+1
8734 IF i8=0 THEN PRINT CODE b$(
u);"
8735 IF i8=1 THEN PRINT #0;CODE
b$(u);"
8736 LET k=k-1
8737 GO TO 8732
8738 IF i8=0 THEN PRINT "
8739 IF i8=1 THEN PRINT #0;"
8740 RETURN

```

```

8749 REM Zmena b$
8750 INPUT "Od bunky=";j
8754 CLS
8756 PRINT "-----beze zmeny,1-k
onec zmeny"
8758 PRINT j;"-";CODE b$(j)
8759 IF j>4000 OR j=0 THEN BEEP
1,20: RETURN
8760 INPUT j$
8762 IF j$="" THEN RETURN
8764 IF j$="n" THEN RETURN
8765 IF j$="t" THEN PRINT: LET j
=j+1: GO TO 8758
8766 LET i=VAL j$
8768 IF i>255 THEN BEEP 1,20: GO
TO 8760
8770 LET b$(j)=CHR$ i
8772 LET j=j+1
8773 PRINT i
8774 GO TO 8756

```

```

8799 REM Prohlizeni
8800 CLS
8802 INPUT "Staticky obr. (a/n)
";k$
8803 LET i8=0
8804 IF k$="n" THEN GO TO 8810
8805 LET i8=1
8806 LET o=1
8808 GO SUB 3000
8810 INPUT "Cislo obr. (0-ukoncu
je) = ";o
8811 IF o=0 THEN LET k$="": RETU
RN
8812 LET s0=0: GO SUB 3000
8814 BEEP 0.1,30: PRINT #0; AT 1,
0;"-----pokrácuje, sprav.odp.=
";o;" PAUSE 0
8815 IF i8=0 THEN CLS: GO TO 88
10
8816 LET i1=1: LET i2=1: LET i3=
0: LET j1=1: LET u=p(o)+3
8817 GO SUB 3700
8818 LET i1=0: LET i2=0
8820 GO TO 8810

```

```

8899 REM Trasovani
8900 CLS
8902 INPUT "Cislo obr (u segment
u +80) = ";o: IF o<1 OR o>100 THEN
BEEP 1,20: RETURN
8903 IF p(o)=0 THEN BEEP 1,20: R
ETURN
8904 LET i8=1
8906 PRINT #0; AT 0,0;"-----"
8908 GO SUB 3000
8910 LET i8=0
8912 RETURN
8914 IF INKEY$<>"k" AND INKEY$<>
"n" AND INKEY$<>"t" THEN GO TO 8
950
8951 BEEP 0.01,40
8952 IF INKEY$="t" THEN LET z=u:
GO SUB 8726: LET u=z
8954 RETURN

```

```

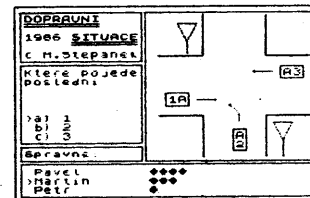
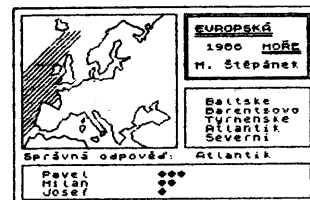
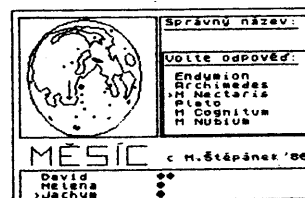
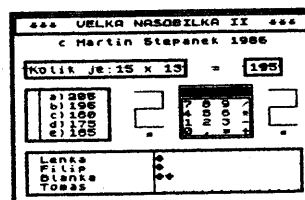
8999 REM Cis vypis b$
9000 CLS
9002 PRINT "Vypis database b$(1):
"
9004 INPUT "Od bunky=";j
9006 FOR a=1 TO 3
9008 FOR i=1 TO 21
9009 IF j+(a-1)*21+i-1>4000 THEN
LET i=22: LET a=5: GO TO 9012
9010 PRINT AT i,(a-1)*10;j+(a-1)
*21+i-1;"-";CODE b$(j+(a-1)*21+
i-1)
9012 NEXT i
9014 NEXT a
9016 RETURN

```

```

9499 REM Tvorba segmentu
9500 LET p0=p0-1
9502 INPUT "Cislo segmentu=";i
PRINT AT i,51+i,1
9503 LET i9=20+i
9504 LET p(i9+1)=b0-2
9506 RETURN
9508 LET j=b0-p(i9)-2
9509 LET b$(p(i9))=CHR$ (INT (j/
256))
9504 LET b$(p(i9)+1)=CHR$ (j-256
*INT (j/256))
9506 RETURN

```



GRAFICKÝ ROZHODOVACÍ SYSTÉM

Ing. Martin Štěpánek

Jordana Jovkova 3256
Praha 4-Modřany

Grafický rozhodovací systém (GROS) je účinný prostředek pro počítačovou podporu rozhodování při výběru nejlepší varianty nebo pro stanovení pořadí výhodnosti variant. Variantou zde může být například způsob řešení problému, druh nějakého zařízení apod. GROS je určen laickému uživateli, takže nevyžaduje žádné znalosti z výpočetní techniky. Komunikace mezi programem GROS a uživatelem je jednoduchá a názorná, protože je vytvořena pomocí grafických prostředků.

SPECTRUM	_____
SPECTRUM128	_____
SPECTRUM128	_____
Apple IIc	_____
Apple IIe	_____
Atari 130XE	_____
Atari 2600ST	_____
Atari 520ST	_____
Atari 800XL	_____
CBM 16	_____
CBM 64	_____
CBM 128	_____
CBM PLUS 4	_____
PC 486	_____
PC 664	_____
PC 8128	_____
SHARP MZ700	_____
Sony Hit Bit	_____

Původní verze programu GROS je napsána v jednoduché verzi jazyka BASIC a je počítáno s možností přenosu programu mezi různými počítači. Nároky na cílový počítač jsou přitom minimální: asi 16 kB operační paměti a práce v reálném čase. Nejvhodnější třídou pro realizaci programu GROS jsou domácí a osobní mikropočítače. Předkládaná verze programu je určena pro počítače ZX-Spectrum.

Způsob řešení problému

Systém umožňuje zadat názvy jednotlivých variant, mezi kterými se budeme rozhodovat. Podobně můžeme zadat názvy všech faktorů, které mají vliv na výběr varianty. K jednotlivým faktorům stanovíme jejich váhy významnosti. Poté pro každou variantu uvedeme míru zastoupení (vlivu) jednotlivých zvolených faktorů. Výsledné ohodnocení zkoumaných variant je dáno vlivem jednotlivých faktorů (podle vah významnosti) zastoupených ve variantách. Přesněji řečeno, ohodnocení je rovno sumě součinů zastoupení faktorů a jejich vah:

$$\text{ohodnocení} = \sum_{\text{faktor}} \text{váha} \cdot \text{zastoupení}$$

Příklad použití systému

Pro snadnější pochopení problematiky nyní uvádím dva příklady různého použití programu GROS.

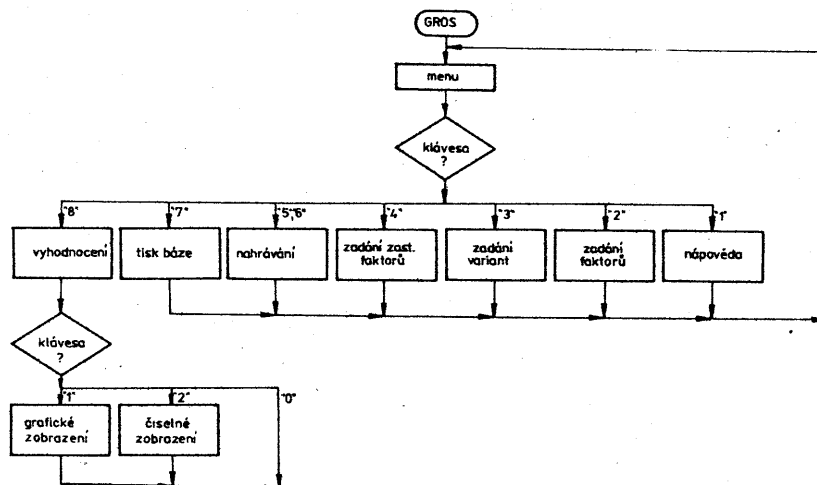
Představme si, že jsme postaveni před problém výběru vhodného typu osobních počítačů pro svůj podnik. Několik typů máme předběžně vybráno (=varianty), ale potřebujeme se rozhodnout jen pro jeden z nich. V první řadě si tedy zvolíme názvy, které budou jednotlivé typy reprezentovat (jsou-li původní názvy krátké, použijeme je, jinak užíváme části názvů nebo zkratky). Dále se zamyslíme nad tím, které vlastnosti počítačů budou ovlivňovat naše rozhodnutí (=ovlivňující faktory). Tyto vlastnosti krátce a výstižně pojmenujeme (např. spolehlivost, servis, cena, rychlost, dosažitelnost atd.). Nyní si

stanovíme, jak mnoho nám na jednotlivých faktorech záleží. Míru důležitosti každého faktoru (=váhu) si v programu GROS můžeme znázornit graficky délkou pásu vedle jména faktoru. Čím je pás delší, tím je faktor důležitější (tj. má větší vliv na rozhodování). Míra důležitosti je též zobrazována v procentech. V poslední fázi se pro každý faktor (tj. vlastnost počítače) vypíše všechny varianty (tedy názvy typů počítačů) a my uvádíme zastoupení příslušného faktoru (vlastnosti) u každé varianty (typu počítače). Jsme-li například požádáni, abchom vyjádřili zastoupení vlastnosti „spolehlivost“, zobrazíme vedle názvu každého typu počítače tím delší pruh, čím větší je jeho spolehlivost. Postupujeme tak, že u nejspolehlivějšího typu zobrazíme plnou délku pruhu a délky ostatních pruhů bereme relativně k ní. Má-li tedy například druhý typ počítače třikrát menší spolehlivost než nejlepší, bude mít třikrát kratší pruh. V této fázi, podobně jako při určování důležitosti faktorů, jistě oceníme grafickou formou zobrazení, kdy jsou na první pohled zřejmé relace mezi zadávanými daty. Po zadání zastoupení vlastností u všech typů počítačů můžeme spustit vyhodnocení, jehož výsledkem je stanovení pořadí míry výhodnosti jednotlivých variant.

V uvedeném příkladu si uživatel vytvářel celou databázi faktů sám. Je však možný i opačný přístup, při kterém je databáze vytvořena odborníkem příslušného oboru a uživatel ji jenom

využívá. Jako příklad tohoto přístupu slouží také demonstrační databáze tohoto programu. Databáze obsahuje vlastnosti různých druhů domácích počítačů a slouží uživateli k výběru počítače na základě jím udaného zastoupení vlastností. Databáze byla vytvořena stejným způsobem jako databáze předcházejícího příkladu. Názvy počítačů jsou chápány jako názvy variant, názvy vlastností jako názvy ovlivňujících faktorů. Zastoupení faktorů ve variantách (tj. vlastností u počítačů) je v bázi zadáno. Databáze má ovšem prázdné pole vyjadřující důležitost faktorů — váhy. Chybějící váhy doplní právě uživatel podle svých požadavků. Jinými slovy, uživatel doplní jak moc mu na jednotlivých vlastnostech záleží. Po doplnění vah může žádat o vyhodnocení, jehož výsledkem je opět žebříček výhodnosti variant. Zde doporučuji důkladné prostudování obrazové přílohy.

Na uvedených příkladech byly ukázány dva různé způsoby používání programu GROS. Cíl obou způsobů je totožný, totiž získání přehledu o vzájemné výhodnosti variant, ale zatímco první možnost předpokládá tvorbu celé databáze uživatelem, v druhém případě se již jedná jen o modifikaci připravené databáze.



Obsluha programu Gros

Po spuštění programu se zobrazí hlavní menu, které nabízí deset možností volby stiskem některé z uvedených číselných kláves. Při stisku klávesy „1“ se například dostaneme do nápovědy, kde je zkráceně uvedeno zhruba totéž co v této kapitole.

Na počátku práce s programem GROS máme v podstatě dvě možnosti: buď budeme vytvářet novou databázi variant, nebo užijeme některou dříve vytvořenou databázi, kterou budeme modifikovat podle aktuálních potřeb. Nahrajeme ji z vnější paměti (např. magnetofon, disketová jednotka apod.)

Při tvorbě nové databáze vstoupíme nejprve do režimu zadávání názvů a vah faktorů. Po stlačení klávesy „2“ se z hlavního menu dostaneme právě do tohoto režimu a uvidíme blikající kurzor. Kurzorem můžeme pohybovat pomocí kurzorových kláves všemi čtyřmi směry. Stisk klávesy odpovídající písmenu, číslici nebo některé jiné značce (kromě dále zmíněných) vyvolá zápis odpovídajícího znaku do pozice kurzoru. Tímto způsobem můžeme zapsat názvy všech faktorů. Když chceme zadat k faktorům jejich váhy, stiskneme klávesu „=“ a dostaneme se do grafického režimu. Míra váhy je dána délkou pásu, kterou lze měnit pomocí kurzorových kláves pro pohyb vlevo a vpravo. Procento odpovídající délce je zobrazováno též číselně ve spodní části obrazovky. Mezi jednotlivými faktory přecházíme užitím kurzorových kláves pro pohyb nahoru a dolů. Zápis vah ukončíme stiskem klávesy odeslání („ENTER“ nebo jí odpovídající klávesa podle typu počítače). Ocitáme se zpět v režimu zadávání názvů. Pokud ho chceme opustit a vrátit se do hlavního menu, stlačíme klávesu „.“. Obvykle zde postupujeme tak, že nejprve zapíšeme názvy všech faktorů (název 1, „ENTER“, název 2, „ENTER“, ...), a pak jim přiřazujeme váhy. Pokud se dopustíme nějaké chyby, stačí kurzorem dojet do příslušné pozice a chybu přepsat (vynechání je přepsání mezerou).

Do režimu názvů variant vstoupíme z hlavního menu stiskem klávesy „3“. V tomto režimu se kurzor opět ovládá kurzorovými klávesami. Znakové klávesy zapisují odpovídající znaky. Klávesou „ENTER“ můžeme přecházet na začátek následujícího řádku. Klávesou „.“ ukončujeme zadávání. Každý název varianty zapisujeme na samostatný řádek.

Posledním zadávacím režimem je popis zastoupení faktorů v jednotlivých variantách. Z hlavního menu se do zmiňovaného režimu dostaneme klávesou „4“. V levé části obrazovky se nám vypíší názvy všech variant, v dolní zóně se objeví název faktoru a na nás je přiřadit zastoupení tohoto faktoru ke všem variantám. Použijeme kurzorové klávesy podobně jako při zadávání vah. Stiskem klávesy „ENTER“ přejdeme

k následujícímu faktoru. Zadávání ukončujeme klávesou „.“. Při vyznačování zastoupení variantě, která má toto zastoupení největší, přiřadíme vždy plnou délku pásu (100%) a ostatní zastoupení posuzujeme relativně k ní (tj. kolikrát je menší)!!!

Do kteréhokoli z uvedených režimů se můžeme později vrátit, abychom opravili chyby nebo provedli modifikaci údajů, které se například časem mohou měnit (zlevnění apod.). Změny provádíme stejnými prostředky jako zadávání. Názvy prostě přepisujeme, hodnoty vyjádřené délkou pásů měníme pomocí kurzorových kláves. Údaje se přijmou vždy v té podobě, v jaké se nacházejí v okamžiku opuštění režimu.

Pomocí kláves „5“ a „6“ se z hlavního menu dostáváme do režimů, které umožňují záznam nebo čtení databáze na vnějším paměťovém médiu. Po

nahrání databáze do počítače můžeme její data modifikovat výše uvedenými metodami.

Režim „7“ umožňuje pořídit výpis databáze metodou kopírování stínítka, na které se postupně vypisují informace z databáze.

Klávesou „8“ v hlavním menu vyvoláváme proces vyhodnocení aktuální databáze. Po vyhodnocení jsme dotázáni, zda si přejeme zobrazení výsledků číselně nebo graficky. Vyhodnocení nám poskytne informaci o ohodnocení jednotlivých variant a to tak, že nejvyšší ohodnocení je bráno jako 100% a ostatní jsou k němu přepočítávána. Abychom neztratili informaci o absolutní výhodnosti, je pod menu výběru způsobu zobrazení výsledků informace o maximální možné hodnotě ohodnocení a o hodnocení nejlepší varianty.

```
Cena
Užitkové p
Systemové p
Hry
Rozsah RAM
Grafika
Barvnost
Klávsnice
Interface
Rychlost

*****
DEMONSTRACNI DATABAZE POCITACU
V tomto režimu uživatel zadá ja
mu záleží na jednotlivých uvede
ných faktorech! Zastoupení fak
toru v počítači je udeláno podle
časopisu "Moj mikro", červenec
1986 (Jugoslavie)
```

Názvy faktorů

```
Spectrum
Spectrum+
Spectrum128
Sinclair QL
Apple IIC
Apple IIE
Atari 130XE
Atari 260ST
Atari 520ST+
Atari 800XL
CBM 16
CBM 64
CBM 128
CBM Plus 4
CPC 464
CPC 664
CPC 6128
SHARP MZ700
Sony HiBit
```

Názvy variant

```
Spectrum
Spectrum+
Spectrum128
Sinclair QL
Apple IIC
Apple IIE
Atari 130XE
Atari 260ST
Atari 520ST
Atari 800XL
CBM 16
CBM 64
CBM 128
CBM Plus 4
CPC 464
CPC 664
CPC 6128
SHARP MZ700
Sony HiBit
```

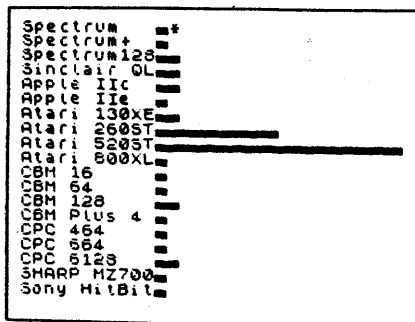
```
Spectrum
Spectrum+
Spectrum128
Sinclair QL
Apple IIC
Apple IIE
Atari 130XE
Atari 260ST
Atari 520ST
Atari 800XL
CBM 16
CBM 64
CBM 128
CBM Plus 4
CPC 464
CPC 664
CPC 6128
SHARP MZ700
Sony HiBit
```

Vybavení uživatelskými programy

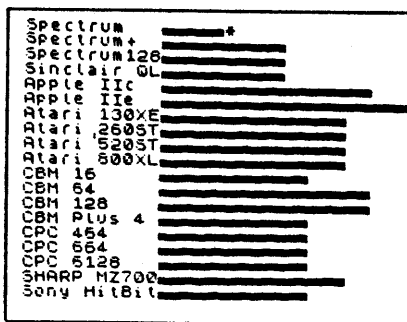
```
Spectrum
Spectrum+
Spectrum128
Sinclair QL
Apple IIC
Apple IIE
Atari 130XE
Atari 260ST
Atari 520ST
Atari 800XL
CBM 16
CBM 64
CBM 128
CBM Plus 4
CPC 464
CPC 664
CPC 6128
SHARP MZ700
Sony HiBit
```

Vybavení systémovými programy

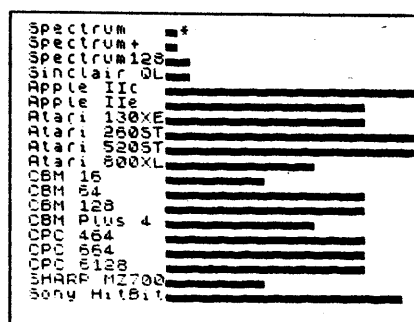
```
Spectrum
Spectrum+
Spectrum128
Sinclair QL
Apple IIC
Apple IIE
Atari 130XE
Atari 260ST
Atari 520ST
Atari 800XL
CBM 16
CBM 64
CBM 128
CBM Plus 4
CPC 464
CPC 664
CPC 6128
SHARP MZ700
Sony HiBit
```



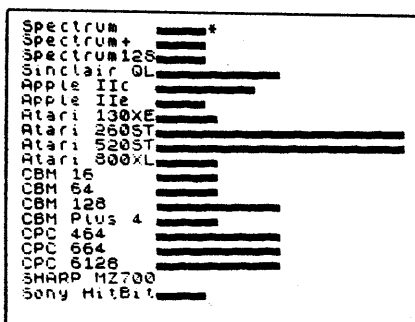
Velikost operační paměti



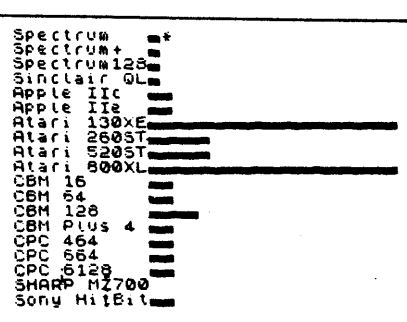
Kvalita klávesnice



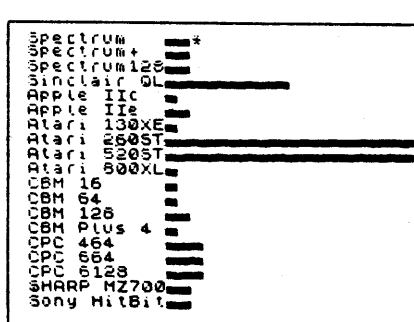
Interfejs v počítači



Jemnost grafiky



Užití barev a možnost výběru



Rychlost procesoru

2. Grafický rozhodovací systém

```
1 GO TO 9000: REM Studeny start
2 GO TO 9090: REM Horky start
```

```
1000 REM FULL SCR EDITOR
```

```
1001 LET x=1
1002 LET r=0
1003 LET f=0
1014 IF k=2 AND f=1 THEN GO SUB 8000
1015 IF k=3 AND f=1 THEN GO SUB 8100
1017 GO SUB 7050
1018 IF f=2 THEN PRINT #0; AT 0,0
INVERSE 1; "NAZVY FAKTORU"
```

```
1019 IF k=3 THEN PRINT #0; AT 0,0
INVERSE 1; "NAZVY VARIANT"
```

```
1020 IF k=2 AND x=2 THEN PRINT #
0; AT 0,0. INVERSE 1; "VAHY FAKTORU"
0; AT 0,0.
vaha="":INT (0.1+(s-91)/90*100);%
```

```
1021 IF k=4 THEN PRINT #0; AT 0,0
INVERSE 1; "FAKTOR:"
0; AT 0,0. "Zastoupe"
```

```
0; AT 0,0. "Zastoupe"
1025 GO SUB 6000
1030 IF c=right THEN GO SUB 7000
GO SUB 6550: GO SUB 7050: GO TO 1499
```

```
1040 IF c=left THEN GO SUB 6500:
GO SUB 7050: GO TO 1499
1050 IF c=down THEN GO SUB 7000:
GO SUB 6000: GO SUB 7050: GO TO 1499
```

```
1060 IF c=up THEN GO SUB 7000:
GO SUB 6050: GO SUB 7050: GO TO 1499
1070 IF x=1 AND c=end AND c=end
ter AND c=graph THEN GO SUB 610
0: GO TO 1499
```

```
1080 IF x=1 AND k=2 AND c=graph
THEN LET x=2: GO SUB 7000: LET s
=v(r+1)+91: GO SUB 7000: GO SUB
7050
```

```
1090 IF x=1 AND c=end THEN GO SUB
61500: RETURN
1100 IF k=2 AND c=center AND k=4
THEN GO SUB 6200: LET x=1: PRINT
#0; AT 0,0; INVERSE 1; "NAZVY FA
KTORU"
GO SUB 7000: LET s=0: GO SUB 7000: GO
SUB 7050
```

```
1110 IF x=2 AND c=center AND k=4
THEN GO SUB 6200: GO SUB 6500
1115 IF k=2 AND c=end THEN GO SUB
6200: LET f=r+1
1120 IF f=70 THEN GO SUB 1500: R
ETURN
```

```
1130 IF x=1 AND c=center THEN GO
SUB 7000: GO SUB 6000: LET s=0:
GO SUB 7000: GO SUB 7050
1499 GO TO 1020
```

```
1500 REM Navrat
```

```
1505 LET f=1
1510 IF k=2 THEN LET f=1
1520 IF k=3 THEN LET f=1
1530 RETURN
```

```
2000 REM EDITOR ZASTOUPENI FAKTO
RU
```

```
2005 LET f=0
2010 LET f=2
2025 GO SUB 6500
2030 GO SUB 1014
2040 RETURN
5000 REM KLAVESA
5020 IF INKEY$="" THEN GO TO 502
0
5030 LET c=CODE INKEY$
5040 BEEP 0.02/20
5050 RETURN
```

```
6000 REM Radek dolu
```

```
6005 IF x=2 THEN GO SUB 6200
6010 LET r=r+1
6020 IF r=70-1 THEN LET r=r-1
6030 IF x=2 AND k=4 THEN LET s=
v(r+1)+91
6035 IF x=2 AND k=4 THEN LET s=z
(r+1)+91
6040 RETURN
```

```
6050 REM Radek nahoru
```

```
6055 IF x=2 THEN GO SUB 6200
6060 LET r=r-1
6070 IF r=0 THEN LET r=0
6075 IF x=2 AND k=4 THEN LET s=
v(r+1)+91
6076 IF x=2 AND k=4 THEN LET s=z
(r+1)+91
6080 RETURN
```

```
6100 REM Zapis znaku
```

```
6110 IF k=2 THEN LET f$(r+1,s+1)
=CHR$ c
6120 IF k=3 THEN LET n$(r+1,s+1)
=CHR$ c
6125 PRINT AT r,s;CHR$ c
6130 GO SUB 6550
6135 GO SUB 7050
6140 RETURN
```

```
6200 REM Odecteni vahy
```

```
6210 IF k=2 THEN LET v(r+1)=s-91
6215 IF k=4 THEN LET z(r+1,f)=s-
91
6220 RETURN
```

```
6500 REM Slopec vlevo
```

```
6504 GO SUB 7000
6505 IF s=s-1 AND x=2 THEN PRINT
#0; AT r,s; " ": LET f$
(r+1,s+1)=" "
6510 LET s=s-1
6515 IF s=0 THEN LET s=0
6530 IF x=2 THEN PRINT AT r,s; " "
```

```
6535 IF x=2 AND k=4 THEN LET f$
(r+1,s+1)=" "
6540 RETURN
```

```
6550 REM Slopec vpravo
```

```
6555 IF x=2 AND s=s+1 THEN PRI
NT AT r,s; " ": LET f$
(r+1,s+1)=" "
6560 IF x=2 AND k=4 AND s=s+1
THEN LET f$(r+1,s+1)=" "
6560 LET s=s+1
6570 IF s=s+1 THEN LET s=s-1
6580 RETURN
```

```
7000 REM Vymaz kurzor
```

```
7010 PRINT AT r,s;k$
7020 RETURN
7050 REM Zobraz kurzor
7055 PRINT AT r,s; " "
7060 IF k=2 THEN LET k=f$(r+1,s
+1): RETURN
7062 IF k=3 THEN LET k=n$(r+1,s
+1): RETURN
7065 LET k$=" ": RETURN
```

```
7100 REM VYHODNOCENI
```

```
7105 CLS
7106 PRINT FLASH 1; "Vyhodnoceni
- prosim cekajte !!!"
7110 FOR i=1 TO r0
7120 LET h(i)=0
7130 FOR j=1 TO r0
7140 LET h(i)=h(i)+v(j)+z(i,j)
7150 NEXT j
7160 PRINT AT 3,0; "Varianta ";i;
" Vyhodnocena:"
7165 BEEP 0.1/10
7170 NEXT i
7175 LET j=0
7180 FOR i=1 TO r0
7185 IF h(i)>j THEN LET j=h(i)
7190 NEXT i
7200 CLS
7205 LET q=j/90
7206 CLS
```

```

7207 PRINT "Zpusob reprezentace
vystedku"
7208 PRINT " 0-Navrat do hl.menu

7209 PRINT " 1-graficky"
7210 PRINT " 2-ciselne"
7211 GO SUB 5000
7212 IF c=CODE "0" THEN LET c=CO
DE "1": RETURN
7213 IF c=CODE "2" THEN GO SUB 7
300 GO TO 7206
7214 IF c=CODE "1" THEN GO SUB 7
215 GO TO 7206

```

```

7215 REM Graficke vyhodnoceni
7216 CLS
7217 GO SUB 8100
7220 FOR i=1 TO r0
7230 FOR j=1 TO s0
7240 IF (j-g1)+q>h(i) OR h(i)=0
THEN LET j=s0+1: GO TO 7250
7245 PRINT AT i-1,j: "="
7250 NEXT j
7260 NEXT i
7265 PRINT #0: AT 1,0: INVERSE 1:
" VYHODNOCENI VYHODNOSTI VARIANT
" GR05 @ Martin STEPANEK 1986

7270 BEEP 0.3,10
7275 GO SUB 5000
7280 RETURN

```

```

7300 REM Ciselne vyhodnoceni
7310 CLS
7320 GO SUB 6100
7330 FOR i=1 TO r0
7340 IF h(i)<0 THEN PRINT AT i-
1,g1+1: INT (h(i)/4/90+100): " %
7350 NEXT i
7355 PRINT #0: AT 1,0: INVERSE 1:
" VYHODNOCENI VYHODNOSTI VARIANT
" GR05 @ Martin STEPANEK 1986

7366 PRINT #0: AT 0,0: INVERSE 1:
" GR05 @ Martin STEPANEK 1986

7367 GO SUB 5000
7370 RETURN

```

```

8000 REM VYPIS NAZVU FAKTORU A V
AM
8010 FOR i=1 TO r0
8020 REM FOR j=1 TO s0
8030 REM PRINT #0(i,j);
8040 PRINT #0(i);
8050 REM NEXT j
8060 NEXT i
8065 BEEP 0.3,10
8070 RETURN

```

```

8100 REM VYPSANI NAZVU VARIANT
8110 FOR i=1 TO r0
8120 REM FOR j=1 TO s0
8125 REM PRINT #0(i,j);
8130 PRINT #0(i);
8140 REM NEXT j
8150 NEXT i
8155 IF k<4 AND k>7 AND k>8 T
HEN BEEP 0.3,10
8170 RETURN

```

```

8500 REM Prechod na dalsi faktor
8510 LET f=f+1
8520 IF f>7 THEN RETURN
8530 CLS
8540 GO SUB 8100
8550 LET s=z(1,f)+g1
8560 LET r=0
8580 GO SUB 7050
8585 REM FOR i=1 TO s0
8586 REM PRINT #1: AT 0,1-1;f;f,
1;
8590 PRINT #0: AT 1,0: INK 7; PAP
ER 3; f;f)
8595 REM NEXT i
8600 FOR i=1 TO r0
8610 FOR q=g1 TO s0
8620 IF q-g1=z(i,f) THEN LET q=
s0+1: GO TO 8654
8653 PRINT AT i-1,q: "="
8654 NEXT q
8656 NEXT i
8657 BEEP 0.3,10
8660 RETURN

```

```

9000 REM INICIALIZACE
9002 REM Uvedeni promenne
nahradte cisly dle uziho pocit
ace
9003 LET graph=CODE "=: LET ent
er=13: LET end=CODE "=:
9004 LET up=11: LET down=10: LET
left=8: LET right=9
9005 LET f=0
9007 LET i=0
9008 LET j=0
9009 LET k=0
9010 REM Max radku editoru
9020 LET r0=22
9021 REM Levy okraj grafiky
9022 LET g1=11
9030 REM Max sloupce editoru

```

```

9040 LET s0=32
9045 LET g0=s0-g1-1
9050 DIM f$(r0,s0)
9055 DIM v$(r0)
9070 DIM n$(r0,s0)
9080 DIM z$(r0,r0)
9085 DIM h$(r0)
9090 REM Telo programu
9091 BORDER 0: INK 7: PAPER 1: C
LS
9092 BEEP 0.1,30
9093 PRINT "
9094 PRINT "
9095 PRINT "
9096 PRINT " @ Martin STEPANEK
K 1986"
9097 PRINT "
9098 PRINT " (verze 1.1
)
9099 PRINT "
9100 PRINT " *** H L A U N I
M E N U ***"
9105 PRINT "
9106 PRINT " 1-Navod pouzivani p
rogramu"
9110 PRINT " 2-Zadani/editace fa
ktoru a vah"
9120 PRINT " 3-Zadani/editace va
riant"
9130 PRINT " 4-Zadani/editace fa
ktoru ve variantach"
9140 PRINT " 5-Ulozeni databaze"
9150 PRINT " 6-Nacteni databaze"
9160 PRINT " 7-Tisk databaze"
9170 PRINT " 8-Vyhodnoceni datab
aze"
9180 PRINT " 9-Zalozeni nove dat
abaze"
9190 PRINT " 0-Ukonceni programu"
9195 PRINT "
9196 PRINT PAPER 2: (Max "r0;"
variant a faktorul)

```

```

9197 REM Ramecek dle scr pocitac
e
9198 PLOT 6,105: DRAW 242,0: DRA
W 0,-92: DRAW -242,0: DRAW 0,92
9200 GO SUB 5000
9210 CLS
9212 LET k=c=CODE "0"
9215 IF c=CODE "1" THEN GO SUB 9
900
9220 IF c=CODE "2" THEN GO SUB 1
000

```

```

9230 IF c=CODE "3" THEN GO SUB 1
000
9240 IF c=CODE "4" THEN GO SUB 2
000
9250 IF c=CODE "5" THEN GO SUB 9
000
9260 IF c=CODE "6" THEN GO SUB 9
000
9270 IF c=CODE "7" THEN GO SUB 9
700
9280 IF c=CODE "8" THEN GO SUB 7
100
9290 IF c=CODE "9" THEN GO TO 1
000
9300 IF c=CODE "0" THEN STOP
9310 GO TO 5090

```

```

9500 REM Vyber pametoveho zarize
ni
9510 CLS
9520 PRINT "Pouzite pametove zar
izeni:
9530 PRINT " 1-Kazetovy magnetof
on"
9540 PRINT " 2-ZX Microdrive"
9550 GO SUB 5000
9560 RETURN

```

```

9700 REM TISK
9710 GO SUB 8000
9720 COPY
9721 BEEP 0.1,30: LPRINT : LPRIN
T : LPRINT
9722 IF INKEY$="" THEN GO TO 972
2
9725 CLS
9730 GO SUB 8100
9740 COPY
9741 BEEP 0.1,30: LPRINT : LPRIN
T : LPRINT
9742 IF INKEY$="" THEN GO TO 974
2
9750 LET f=0
9755 IF f+(i+1,1)=0 THEN LET f
=f+1: GO TO 9760
9760 GO SUB 8500
9770 COPY
9771 BEEP 0.1,30: LPRINT : LPRIN
T : LPRINT
9772 IF INKEY$="" THEN GO TO 977
2
9780 IF f<0-1 THEN GO TO 9755
9790 RETURN

```

```

9800 REM SAVE
9810 CLS
9811 PRINT "Nazev databaze?"
9812 INPUT i$
9813 IF LEN i$>7 THEN PRINT "Dov
oleno max 7 znaku!": GO TO 9820
9815 GO SUB 9500
9816 IF c=CODE "1" THEN GO SUB 9
820
9817 IF c=CODE "2" THEN GO SUB 9
834
9818 RETURN

```

```

9820 REM Save na m9f
9825 SAVE i$+"1" DATA f$(i)

```

```

9826 SAVE i$+"2" DATA v(i)
9827 SAVE i$+"3" DATA n$(i)
9828 SAVE i$+"4" DATA z(i)
9829 RETURN
9834 REM Save na microdrive
9835 SAVE "m";1;i$+"1" DATA f$(i)
9836 SAVE "m";1;i$+"2" DATA v(i)
9837 SAVE "m";1;i$+"3" DATA n$(i)
9838 SAVE "m";1;i$+"4" DATA z(i)
9840 RETURN

```

```

9850 REM LOAD
9851 CLS
9852 PRINT "Nazev databaze?"
9853 IF LEN i$>7 THEN PRINT "Dov
oleno max 7 znaku!": GO TO 9870
9854 GO SUB 9500
9855 IF c=CODE "1" THEN GO SUB 9
860
9856 IF c=CODE "2" THEN GO SUB 9
870
9858 RETURN

```

```

9860 REM Load z m9f
9864 LOAD i$+"1" DATA f$(i)
9865 LOAD i$+"2" DATA v(i)
9866 LOAD i$+"3" DATA n$(i)
9867 LOAD i$+"4" DATA z(i)
9868 RETURN

```

```

9870 REM Load z microdrive
9874 LOAD "m";1;i$+"1" DATA f$(i)
9875 LOAD "m";1;i$+"2" DATA v(i)
9876 LOAD "m";1;i$+"3" DATA n$(i)
9877 LOAD "m";1;i$+"4" DATA z(i)
9890 RETURN

```

```

9900 REM HELP PRO ZX SPECTRUM
9901 LET i$="STISKNETE LIBOVOLNO
U KLAUESU !!!"
9902 CLS
9904 PRINT " GR05 slouzi pro roz
hodovani me-z vice moznymi vari
antami prese-ni nebo pro stanoven
i poradí a vřřvř vyhodnosti jedn
otlivých va-riant.
9905 PRINT "
9906 PRINT " GR05 umožňuje pro k
azdy ovřřvř vřřvřící faktor zvolit
i významovou vahy a pro každou va
riantu určit zastoupení jednotliv
ých faktorů. Ustředně ohodnocení
každé uvede-ne varianty je dano
součtem sou-činu jednotlivých va
h s přísluš-ným zastoupením fakt
oru.
9907 PRINT "
9908 PRINT " ohodnocení=SUMA(va
ha*zastoup.)
9909 PRINT "
9910 PRINT " Popis ovládání v ne
kterých re- zimech je na dalších
stránkách. U ostatních případec
h se řídíme podle nabídkových me
nu.
9911 PRINT #0: INVERSE 1; i$
9912 GO SUB 5000
9913 CLS
9914 PRINT "ZADANI NAZVU FAKTORU
A VAH:"
9915 PRINT "
9916 PRINT " a)Kurzor:5,6,7,8 ne
bo kurzorové klávesy"
9918 PRINT " b)Prechod na další
radku: ENTER"
9920 PRINT " c)Režim vah: a"
9922 PRINT " d)Zpet režim nazvu:
ENTER"
9924 PRINT " e)Ukončení:."
9925 PRINT "
9926 PRINT "Pozn.:Názvy faktorů
musí být od prvního sloupce, jin
ak se nevy- tisknou! Max délka n
ázvu je zde
9927 PRINT #1: " znaku."
9928 PRINT #0: INVERSE 1; i$
9930 GO SUB 5000
9932 CLS
9934 PRINT "ZADANI NAZVU VAH:"
9935 PRINT "
9936 PRINT " a)Kurzor:5,6,7,8 ne
bo kurzorové klávesy"
9937 PRINT " b)Prechod na další
radku: ENTER"
9938 PRINT " c)Ukončení:."
9939 PRINT "
9940 PRINT "Pozn.Max délka názvu
je zde
9942 PRINT #0: INVERSE 1; i$
9944 GO SUB 5000
9946 CLS
9948 PRINT "ZADANI FAKTORU VE VA
RIANTACH:"
9949 PRINT "
9950 PRINT " a)Kurzor:5,6,7,8 ne
bo kurzorové klávesy"
9952 PRINT " b)Prechod k dal. fa
ktoru: ENTER"
9954 PRINT " c)Ukončení:."
9956 PRINT #0: INVERSE 1; i$
9958 GO SUB 5000
9960 CLS
9962 PRINT "
9964 PRINT FLASH 1: "UVEDENY NAVO
D PLATI BEZ VYJIMEK JEN PRO POCI
TAC ZX SPECTRUM !!!
9966 PRINT "
9968 PRINT "Tento navod je pouze
orientacni,proto je vhodne si p
recist pri- rucku uzivatele."
9970 PRINT #0: INVERSE 1; i$
9972 GO SUB 5000
9974 RETURN

```


MĚSTA

Jan Jelínek

Kaplická 852
140 00 Praha 4

Výukový program MĚSTA slouží k procvičování (případně testování) znalosti polohy měst nad 10 000 obyvatel v ČSSR.

Je rozdělen do dvou částí:

1. Vstupní test: hráč má za úkol určit název počítačem označeného města na slepé mapě ČSSR. V této části se vyskytují pouze města nad 50 000 obyvatel a není bodována. Určí-li hráč správně alespoň tři města z pěti, postupuje do druhé, hlavní části.
2. Hlavní část: v této části jsou postupně probírány jednotlivé kraje. Úkolem hráče je určit zaměřovačem polohu města, jehož název zadá počítač (10 000 až 50 000 obyvatel). V každém kraji jsou města zadávána tak dlouho, dokud součet odchylek určených poloh od skutečnosti nepřesáhne 100 km (maximálně však 5krát). O kolik se může hráč ještě zmylit je zobrazeno jako dovolená chyba a pokud není překročena, připočítá se za každé město 100 bodů. Nevyužije-li hráč během 5 pokusů dovolené chyby, je mu za každý zbylý kilometr připočítán bod.

V jednom kraji lze získat maximálně 600 bodů. Pro větší obtížnost zadává někdy počítač i města z jiného kraje než zkoušeného. Po probírání všech 10 krajů vyhodnotí počítač výsledek.

Zobrazení měst: v 1. částiv 2. části
nad 1 000 000 ○ ○
100 000—1 000 000 ○ ○
50 000— 100 000 + o
25 000— 50 000 +
10 000— 25 000 +
(stav k 1. 4. 1982)

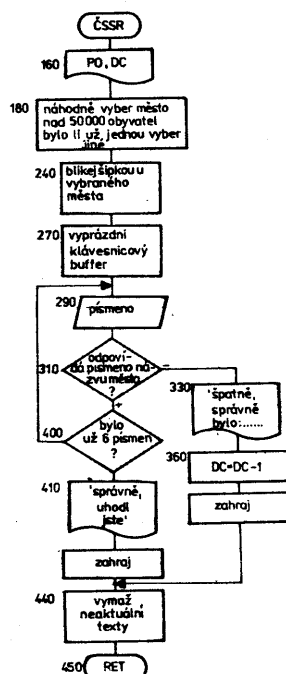
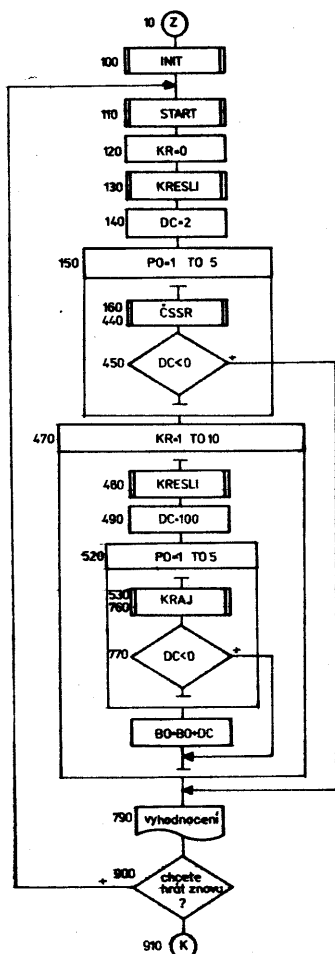
Obsluha programu

Program nahrajeme příkazem CHAIN (skládá se ze tří záznamů). Po vykreslení obrysů republiky začne blikat u některého města šipka. Napíšeme název tohoto města (bez háčků a čárek). Počítač kontroluje pouze prvních šest písmen a na chybu reaguje již při prvním nesprávném písmenu.

V druhé části ovládáme zaměřovač (blikající křížek) pomocí šipek. Po umístění zaměřovače na žádané místo stiskneme tlačítko „1“. Myslíme si, že zadané město není v příslušném kraji, stiskneme tlačítko „1“. Myslíme-li si, že měřovače nezáleží). Nesprávné určení kraje se hodnotí jako chyba 50 km. Nebyla-li odpověď přesná, označí blikající šipka správné místo.

Popis některých podprogramů

INIT — inicializuje pole
Q(10,2) — souřadnice map krajů
NŠ(10) — názvy krajů
TŠ(6) — texty vyhodnocení
— načte z mgf pole



X(203) — x souřadnice měst
Y(203) — y souřadnice měst
O(203) — počty obyvatel ve městech
H(444) — zakódované hranice ČSSR
K(338) — zakódované hranice krajů
MŠ(203) — názvy měst
— načte z mgf tvary miniznaků (64 na řádek)

(tisk miniznaků pomocí GOSUB \$M1)

START — vynuluje pole Z(203)
— příznaky, zda město bylo již jednou vybráno

KRESLI — spustí hraní hudby (GOSUB \$M3)

— nakreslí obrysy kraje nebo republiky (GOSUB \$M2)

— nakreslí příslušná města

— nakreslí rámeček s nápisy

— počká, až dohraje hudba

DIST — vyhodnotí odpověď hráče:

— zmylil-li se v kraji, pak DI=50,

— jinak DI = vzdálenost zaměřovače od města v km,

— je-li DI<4, pak DI=0.

KEY

— je-li stisknuto tlačítko se šipkou,

pohybuje zaměřovačem v jejím směru,

— stiskne-li se „1“, vrátí se na STRIL,

— stiskne-li se „2“, vrátí se na MIM.

Některé proměnné:

KR — číslo kraje (0=ČSSR),

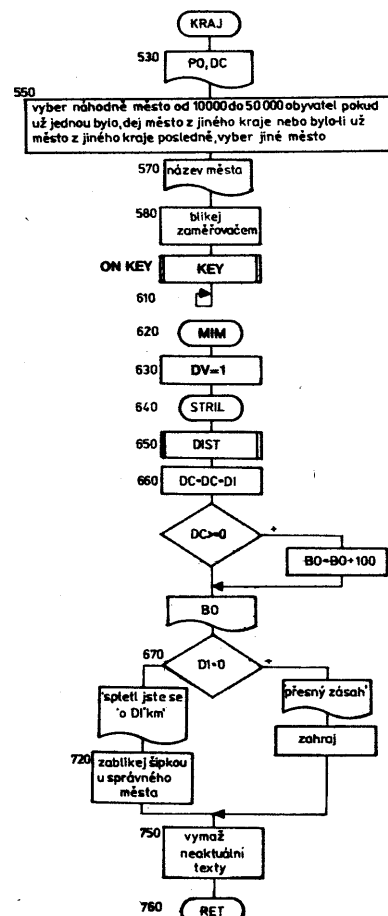
DC — dovolená chyba (v 1. části počet omylů, v 2. části počet kilometrů),

BO — počet bodů,

PO — číslo pokusu,

DI — vzdálenost zaměřovače od města,

DV — příznak, že bylo stisknuto tlačítko „2“.



Výpis programu „Města“

(Výpis neobsahuje data o městech, tj. zakódované hranice krajů a republiky, souřadnice a názvy měst a počty obyvatel. Těchto dat by bylo asi 8 str. A4. Výpis je určen pro pochopení problematiky a ukázkou programovací techniky. Program byl sestaven pro počítač SORD M5 s BASIC-G a pamětí EM-32).

```

10 rem (c) jan jelinek ml. 1986
100 gosub #init
110 gosub #start
120 console 1:kr=0
130 gosub #kresli
140 dc=2
150 for po=1 to 5
160 locate 22,2:m1#num$(po):gosub #m1
170 locate 26,3:m1#num$(dc)+' krat':gosub #m1
180 m=rnd(31):me=-1
190 repeat
200 me=me+1
210 if o(me)>=500 then m=m-1
220 until m<0
230 if z(me)then goto 180 else z(me)=1
240 loc 1 to (x(me)-m2x)/2+2,(y(me)-m2y)/2+2
250 on event gosub #sip
260 event 25:event on
270 repeat:until inkey#=""
280 m=len(m$(me)):n=0:if m>6 then m=6
290 n=n+1
300 r=asci$(inkey#):if r=-1 then goto 300 else p=r and 31
310 c=asci$(mid$(m$(me),n))and 31
320 if p=c or p=5 and c=6 or p=21 and c=22 then goto 380
330 event off:scol 1,0:play 'rge-cge-c':locate 18,5:m1#left$(m$(me),n-1)+chr$(r):gosub #m1
340 locate 18,6:m1#='spravnf bylo':gosub #m1
350 locate 18,7:m1#m$(me):gosub #m1
360 dc=dc-1
370 goto 420
380 if n=m then n=18
390 locate 18,5:m1#left$(m$(me),n):gosub #m1
400 if n=m then goto 290
410 event off:scol 1,0:play 'cegcegc2':locate 18,6:m1#='spravnf,
uhodl jste':gosub #m1
420 for n=1 to 10000:next
430 locate 18,5:m1#rpt$(26,' '):gosub #m1
440 locate 18,6:gosub #m1:locate 18,7:gosub #m1
450 if dc>=0 then next po else goto #konec
460 loc 0 to 128,96
470 console 0:for kr=1 to 10
480 gosub #kresli
490 dc=100
500 zm=q(kr-1,0):repeat:zm=zm+1:until o(zm)<500
510 km=q(kr,0)-1
520 for po=1 to 5
530 locate 22,2:m1#num$(po):gosub #m1
540 locate 26,3:m1#num$(dc)+' km ':gosub #m1
550 me=rnd(km-zm)+zm
560 if z(me)then if mi then goto 550 else me=me+100+200*(me>100)
570 locate 18,5:m1#m$(me):gosub #m1
580 dv=0:on event gosub #kriz
590 on key gosub #key
600 event 25:event on:key on
610 goto 610
620 #mim
630 dv=1
640 #stril
650 gosub #dist
660 dc=dc-d1:if dc>=0 then bo=bo+100
670 if d1 then m1#='spletl jste se o '+num$(d1)+' km' else m1#='
presny zasah/'
680 locate 18,6:gosub #m1
690 locate 27,2:m1#num$(bo):gosub #m1
700 loc 1 to sx+1,sy+1
710 on event gosub #sip
720 if d1 then event on else play 'cegcegc2'
730 for n=1 to 10000:next
740 event off:scol 0,0:scol 1,0
750 locate 18,5:m1#rpt$(26,' '):gosub #m1
760 locate 18,6:gosub #m1
770 if dc>=0 then next po:bo=bo+dc
780 next kr
790 #konec
800 fcol 7:locate 0,11:gmode 0
810 for n=0 to 5
820 print#1 rpt$(26,' ')
830 next n
840 box 0,88,207,135:locate 1,12
850 m1#='dosahl jste '+num$(bo)+' bodv z 6000 moznych':gosub #m1
860 print 'u':m1#='dnesni rekord: '+num$(re)+' bodv':gosub #m1
870 print 'u':m1#='(bo/1000):gosub #m1
880 print 'u':m1#='chcete hrát znovu? (a/n)':gosub #m1
890 if bo>re then re=bo
900 if inkey(0)=0 then goto 900
910 if inkey(0)=25 then goto 110 else print 'x':stchr:console 1:
end
920 #dist
930 gmove#0:di=0
940 if m1 then sx=152:sy=64 else sx=x(me)-m2x:sy=y(me)-m2y
950 if dv then mi=mi+1
960 if mi=1 then di=50
970 if m1 then return
980 d2=dist(sxx/2,sy, pos(2)*3/2, pos(3)):if d2=-1 then di=200:re
turn
990 repeat:di=di+1:until di>=d2
1000 if di<4 then di=0
1010 return
1020 #sip
1030 scol 1,sprite(1,3)xor 7
1040 play 'g8'
1050 return
1060 #kriz
1070 scol 0,sprite(0,3)xor 7
1080 play 'c32'
1090 return
1100 #key
1110 if inkey(0)then goto 1150
1120 move 0 on#0
1130 event on
1140 return
1150 event off:scol 0,7
1160 if inkey(0)=9 then key off:return #stril
1170 if inkey(0)=10 then key off:return #mim
1180 if inkey(0)=55 then move 0 step 1,0,2
1190 if inkey(0)=54 then move 0 step -1,0,2
1200 if inkey(0)=46 then move 0 step 0,1,2
1210 if inkey(0)=51 then move 0 step 0,-1,2
1220 return
1230 #kresli
1240 ginit:gmode 4:gosub #m3
1250 scod 0,139:scol 0,0:scod 1,140:scol 1,0
1260 m2m=1-(kr=0):m2x=q(kr,1):m2y=q(kr,2)
1270 m2=varptr(h(0)):fcol 6:gosub #m2
1280 if kr then goto 1350
1290 fcol 11
1300 for n=0 to 203
1310 if o(n)<500 then goto 1330
1320 gmove(x(n)-m2x)/2,(y(n)-m2y)/2:circle 1-(o(n)>1000)-(o(n)>1
0000)
1330 next n
1340 goto 1400
1350 m2=varptr(k(0)):fcol 12:gosub #m2
1360 fcol 11
1370 for n=q(kr-1,0)to q(kr,0)-1
1380 gmove x(n)-m2x,y(n)-m2y:circle-(o(n)>250)-(o(n)>500)-(o(n)>
1000)-(o(n)>10000)
1390 next n
1400 gmode 0:for n=0 to 8:print#1 cursor(17,n)rpt$(15,' '):next
1410 fcol 5:box 136,0,255,72
1420 locate 18,2:m1#='pokus: body: '+num$(bo):gosub #m1
1430 locate 18,3:m1#='dovolena chyba':gosub #m1
1440 if kr then goto 1480

```

```

1450 locate 18,1:m1$='ceskoslovensko':gosub #m1
1460 locate 18,4:m1$='jake je to m1sto?':gosub #m1
1470 goto 1510
1480 locate 18,1:m1$='kraj: '+num$(kr)+' - '+n$(kr):gosub #m1
1490 locate 18,4:m1$='hledajte m1sto:(potom '1')':gosub #m1
1500 locate 18,7:m1$='2' = neni v tomto kraji':gosub #m1
1510 fool 9
1520 if play(0)then goto 1520
1530 return
1540 init
1550 randomize:re=0
1560 print 'u1vsl':;len 18
1570 dim q(10,2),n$(10),z(203)
1580 for n=0 to 10:for m=0 to 2
1590 read q(n,m)
1600 next m,n
1610 for n=1 to 10:read n$(n):next
1620 old 'datarepub'
1630 dim m$(203),x(203),y(203),o(203),h(444),k(338)
1640 pokew #7264,4080,varptr(m$(0)),0:call #0e7d
1650 pokew #7264,408,varptr(x(0)),4080:call #0e7d
1660 pokew #7264,408,varptr(y(0)),4488:call #0e7d
1670 pokew #7264,408,varptr(o(0)),4896:call #0e7d
1680 pokew #7264,890,varptr(h(0)),5304:call #0e7d
1690 pokew #7264,678,varptr(k(0)),6194:call #0e7d
1700 len 50:dim t$(6):m1$=''
1710 for n=0 to 6:read t$(n):next
1720 print 'vrv'
1730 old 'm51minizn'
1740 stchr 'f0c0a09008040201' to 140
1750 return
1760 #start
1770 bo=0
1780 for n=0 to 203:z(n)=0:next n
1790 print 'u';
1800 return
1810 data 0,8,-33,19,58,10,28,58,98,41,0,54,63,37,0
1820 data 83,138,14,107,138,96,140,204,34,167,212,182,190,270,13
0,204,351,112
1830 data stredocesky,jihocesky,zapadocesky,severocesky,vychodoc
esky,jihomoravsky,severomoravsky
1840 data zapadoslovensky,stredoslovensky,vychodoslov.
1850 data je to hrva - nemate ani zakladni prehled/,za vas vyko
n by se stydli i simpanz/
1860 data vas vykon je velmi slaby - bfzte si odpocinout/,neni

```

```

to nejhors1, ale zkuste se zlepšit/'
1870 data 'mate velmi dobry prehled, jen tak dal/',jste az moc c
hytry (anebo svindlujete)/
1880 data to je podvod - s vami nechci hrát/
30000 #m1
30010 gmode 0:for m1=1 to len(m1$)
30020 print #1 chr$(ascii(mid$(m1$,m1,1))+128*(m1 mod 2));
30030 if m1 mod 2 then print '5';:gmode 4 else gmode 0
30040 next:gmode 4
30050 return
30060 #m2
30070 poke #7120,#fe,0,#d20,2,#7e,#d23,#fe,3,#dc8,#cd,#4c,#d71,#dc5,#d
d5,#c1,#d1,#cd,#4c,#d71,#dc5,#d5,#c1,#d1,#dc9-44*(m2m-1)
30080 poke #7138,#78,#fe,0,#d20,#0c,#7a,#fe,0,#d20,7,#7b,#fe,#c0,#d
30,2,#f1,#c9,#f1,#d18,#d4
30090 poke #714c,#f5,#e6,3,#fe,3,#d20,1,3,#fe,1,#d20,1,#0b,#f1,#cb
,#3f,#cb,#3f,#c9
30100 if peekw(m2)=-1 then return
30110 pokew #7262,0,peekw(m2)-m2x,peekw(m2+2)-m2y,m2+4
30120 call #7120
30130 plot peekw(#7264)/m2m,peekw(#7266)/m2m
30140 if peek(#7263)=3 then m2=peekw(#7268):goto 30100 else goto
30120
30150 #m3
30160 len 32:restore 30290:m3=0
30170 on event gosub 30200
30180 event 30: event on
30190 return
30200 read m3$:if m3$='a' then goto 30270
30210 m3=m3+1
30220 if val(m3$)then m3n=m3:repeat:read m3$:until m3$='#'+m3$
30230 if m3$='x' then restore 30290:m3=0:repeat:read m3$:m3=m3+1
until m3=m3n
30240 if m3$='.' then event off:return
30250 if left$(m3$,1)='<' then goto 30200
30260 event val(mid$(m3$,2)):goto 30200
30270 m3=m3+3:play m3$:read m3$:play,m3$:read m3$:play,m3$
30280 return
30290 data 05L4s1t215,05L4s1,05L1s1,1,1,2,3,#55,b-agg,d2.04go5,f
,4,5,6,7,7,2,b-ag2,c1,e
30300 data 4,b-ag2,d2.04go5,f,5,c2c2,r1,r,.
30310 data #1,b-agr,c2.04go5,e,#2,b-agg,c2.04go5,e,#
30320 data #3,b-agr,c1,e,#4,b-agr,d2.04go5,f,#
30330 data #5,gge-d,fo4gabo5,r,#6,cr2.,co4bo5ce,r,#
30340 data #7,b-ag2,c2.04go5,e,#

```

SLOVNÍK

M. Kaftan
Neratovická 32
182 00 Praha 8

Program Slovník slouží v šesti základních způsobech použití k ukládání, uchování, automatickému řazení, vyvolání a případnému vytištění slov v kterémkoli cizím jazyce. Jednoduché úkony jsou voleny tak, aby Slovník optimálně plnil svůj účel jako praktická a rychlá doplnková jazyková pomůcka.

Slovník je určen pro používání na osobním počítači ZX Spectrum s vhodnou kompatibilní tiskárnou. Základní verze programu je přizpůsobena pro tisk prostřednictvím Interface 1.

- K jeho hlavním výhodám patří:
- možnost vlastního zápisu slov s jejich okamžitým samočinným abecedním řazením,
 - rozsah do 800 použitelných slov,
 - vyvolání celých stránek v počtu 18 výrazů nebo jednotlivých slov,
 - možnost rychlých vlastních úprav (vymazávání, doplňování, opravy textu) podle potřeby
 - aplikace na magnetofonových páskách či microdrivu a vhodné tiskárně.

Výklad programu

1. Základní výběr (menu)

Program Slovník umožňuje variabilní využití těchto základních módů:

1. Zápis nových výrazů...NOVÉ SLOVO
 2. Změna dvojice výrazů na jejich opačné pořadí.....ZMĚNA VERZE
 3. Vyvolání zapsaných slov na displej.....ČTENÍ SLOVNÍKU
 4. Nahrání do počítače/uchování záznamu.....LOAD/SAVE
 5. Použití tiskárny.....VYTIŠTĚNÍ
 6. Zkoušení jednotlivých slovíček.....ZKOUŠENÍ SLOV
- Další část je zaměřena na podrobnější popis výše uvedených variant funkce.

Základní výběr (menu) je východiskem každého jednotlivého módu a následných operací. Objeví se automaticky na konci každé nahrávky po povelu „LOAD“.

1. Zápis nových výrazů — NOVÉ SLOVO

Stiskneme-li ze základního výběru tlačítko 1, objeví se nejdříve výzva k potvrzení. Zmačkne-li dále A (ANO), objeví se „stránka“ slovníku, připravená k zapisování. Začátek zápisu označuje žlutý kurzor, který se při psaní automaticky posouvá. Stránka je rozdělena na dvě poloviny, z nichž každá má kapacitu 16 znaků. Počet řádků na každé stránce je 18.

a) zapisování, pohyb kursoru
Pohyb kursoru se ovládá příslušnými šipkami. Přejít z levé poloviny „stránky“ na pravou se provádí tlačítkem ENTER. Tímž tlačítkem se převádí kurzor po ukončení v pravé polovině stránky znovu na začátek její levé poloviny.

b) přepisování, vymazávání
— „překlep“, tj. chybu při psaní lze

jednoduše opravit tak, že se kurzorem vrátíme zpět k poslednímu správnému písmenu a odtud začneme psát správný text,

- stejný účinek má tlačítko DELETE, které současně vymazává,
- celý řádek lze vymazat současným zmáčknutím SYMBOL SHIFT a S (NOT). Viz též odst. A/II
- pomocné tlačítko EDIT.

c) značení počtu slov a číslování stránek

Při zápisech se číslice objevují automaticky na spodním okraji displeje.

2. Změna verze

Změna verze se volí ze základního výběru tlačítkem 2. Výsledkem je záměna obou polovin na dotyčné stránce. Pokud je např. slovník organizován tak, že vlevo jsou zapisovány výrazy v cizím jazyce a vpravo jejich české ekvivalenty, po stisknutí tlačítka se jejich poloha zamění. Dochází rovněž automaticky k přechodu do módu 3, tj. čtení slovníku.

3. Vyvolání zapsaných slov na displej — ČTENÍ SLOVNÍKU

Stiskneme-li ze základního výběru tlačítko 3, objeví se po následném A (ANO) dosavadní zápisy počínaje stránkou č. 1. Další stránky se vyvolají tlačítkem EDIT.

I v módu č. 3 lze provádět dodatečné zápisy. Cursor se po celé ploše stránky posunuje pomocí tlačítek se šipkami.

4. Operace LOAD a SAVE

- Po tlačítku 4 a A (ANO) se nejdříve objeví výběr LOAD/SAVE k volbě tlačítka L (LOAD) nebo S (SAVE).

a) nahrání záznamů do počítače — LOAD

Stisknutí L dává alternativu použití magnetofonové pásky (tlačítko 0) nebo jednoho z 8 možných microdrive (tlačítko 1 až 8).

b) uchování záznamu — SAVE

Postup SAVE je obdobný. Po volbě záznamové jednotky 0 až 8 se objeví nápověď „Název“, pod níž má být záznam uchován a který dopíšeme. Po stisknutí ENTER se nahrávají na pásku uložená slova, po čemž se v případě nahrávání na microdrive slova automaticky verifikují. Jestliže se nahrávají na pásku, pak lze verifikovat stisknutím „a“ po objevení „Verify“.

5. Vytisknutí

Po stisknutí tlačítka 5 následuje nejdříve upozornění, že program umožňuje použití tiskárny pouze prostřednictvím ZX Interface 1. Rozsah tištěného textu je od začátečního do koncového slova.

6. Zkoušení slov

Zkoušení (tlačítko 6) spočívá v udání (vepsání) přesného překladu k výrazům, které počítač náhodně vybírá ze slovníku. Postačí k tomu stisknutí kteréhokoli tlačítka po nápovědi „Zkoušet?“, po níž se napíše příslušný překlad (ekvivalent) k výrazu, který se objeví v rubrice „Slovo“.

Výpis programu „Slovník“

```

0 REM *****
***** SLOVNÍK ***
****
* MARTIN KAFTAN 1986 **
*****
1000 REM Inicializace*****
1001 CLEAR 65500: DIM I$(800,32): DIM d$(6,15): POKE 23609,50: LET f=0: LET poc=
0: LET n$="" STOP...menu EDIT...instrukce ": LET m$=CHR$ 16+CHR$ 7+CHR$ 17+CHR$
1+CHR$ 19+CHR$ 1: LET u$=CHR$ 16+CHR$ 0+CHR$ 17+CHR$ 6
1002 RESTORE 1026: FOR p=1 TO 6: BEEP .01,30+p: READ d$(p): NEXT p

1003 RESTORE 1116: FOR p=1 TO 28: READ d: POKE 65499+p,d: NEXT p: GO TO 1021

1004 REM Razeni podle abecedy***
1005 FOR p=poc-1 TO 2 STEP -1: LET sd=0: FOR c=poc-1 TO poc-p STEP -1
1006 IF I$(c+1)+I$(c) THEN LET f$=I$(c): LET I$(c)=I$(c+1): LET I$(c+1)=f$: LET
sd=sd+1
1007 NEXT c: LET p=p*(sd+0): NEXT p
1008 RETURN
1009 REM Vyhledavani slova*****
1010 INPUT "Hledane slovo:"; LINE Z$: IF Z$="" OR Z$=" " THEN BEEP .2,20: GO TO
1010
1011 LET dl=LEN Z$: LET lo=2: LET po=poc: LET q=2
1012 IF I$(po, TO dl)=Z$ THEN LET q=po: GO TO 1019
1013 IF I$(lo, TO dl)=Z$ THEN LET q=lo: GO TO 1019
1014 LET ps=INT ((po+lo)/2)
1015 IF I$(ps, TO dl)=Z$ THEN LET q=ps: GO TO 1019
1016 IF I$(ps)+Z$ THEN LET lo=ps
1017 IF I$(ps)+Z$ THEN LET po=ps
1018 IF po-lo=1 THEN GO TO 1014
1019 LET yo=INT (q/20): LET yo=yo+18+1: LET y=q-yo+(q-yo): GO TO 1058
1020 REM Menu*****
1021 BORDER 1: PAPER 1: INK 7: CLS : PRINT INVERSE 1;AT 0,11;TAB 20;AT 1,11;" SL
OVNIK ";AT 2,11;TAB 20: PRINT AT 5,0: FOR p=1 TO 6: PRINT TAB 7,p;"...";d$(p):
PRINT : NEXT p
1022 BEEP .1,20: PRINT AT 21,0;TAB 5;"Stiskni prislusne cislo";TAB 31: PAUSE NOT
P1: LET c=PEEK 23560: LET w$=CHR$ c: IF w$="1" OR w$="6" THEN BEEP .1,1: GO TO
1022
1023 BEEP .1,30: PRINT u$;AT VAL "21",VAL "0";TAB VAL "8";" Potvrd (ano) "; FLAS
H 1,w$: FLASH 0;TAB 31: PAUSE 0: IF INKEY$<"a" AND INKEY$>"A" THEN GO TO 1022

1024 REM Vetveni*****
1025 BORDER 7: PAPER 5: INK 0: GO TO 1000+27*(w$="1" OR w$="3")+59*(w$="2")+61*(
w$="4")+79*(w$="5")+97*(w$="6")
1026 DATA "Nove slovo","Zmena verze","Cteni slovníku","Load/Save","Vytisteni","Z
kouseni slov"
1027 REM Nove slovo a ***** Cteni slovníku*****
1028 LET ct=(0 AND w$="1")+1 AND w$="3": LET x=0: LET ox=X: LET No=X
1029 CLS : IF w$="3" THEN LET yo=1: LET y=1: LET oy=y: GO TO 1058
1030 LET yo=INT (poc/18): LET yo=yo+18: IF yo<1 THEN LET yo=1
1031 LET y=poc-yo+2: LET oy=y: GO TO 1058
1032 PRINT AT 0,0;m$;" ";TAB 32;TAB (INT (33-LEN d$(VAL w$))/2);d$(VAL w$);TAB 3
2:" ";TAB 32
1033 POKE 23659,0: PRINT PAPER 4;AT 22,0;" Pocet slov:";poc;" strana c.:";INT
(yo/18)+1;TAB 32;n$: POKE 23659,2
1034 REM Pohyb a psani na ***** obrazovku*****
1035 BEEP .003,40: PRINT AT oy+2,ox+xo;I$(oy+yo,ox+xo+1);AT y+2,x+xo: PAPER 6;1
$(y+yo,x+xo+1): LET oy=y: LET ox=x
1036 PAUSE 0: LET c=PEEK 23560: LET a$=CHR$ c: IF c/31 AND c/128 THEN LET I$(y+y
o,x+xo+1)=a$: LET x=x+1
1037 IF ox/15 AND c/31 AND c/128 THEN GO TO 1035
1038 IF ct THEN GO TO 1045
1039 IF x/15 AND (c/12)+(c/13)<0 THEN LET x=x-1: BEEP .1,40: GO TO 1035
1040 IF c/13 AND I$(y+yo,xo+1)=" " THEN BEEP .1,20: GO TO 1036
1041 IF c/13 THEN POKE 22592+x*xo+32*y,40: LET y=y+(xo+16): LET poc=poc+(xo+16):
LET xo=xo+(xo+16)+16*(xo=0): LET ox=0: LET x=0
1042 IF c/13 AND xo=0 AND poc+1 THEN GO SUB 1005: GO TO (1056 AND y/18)+(1058 AN
D y/18)
1043 IF c/13 AND xo=0 AND poc+2 THEN GO TO 1033
1044 IF (c/8)+(c/9)=0 THEN LET x=x-(c/8)+(c/9): LET x=x+(xo)-(x/15): GO TO 1035
1045 IF c/12 THEN PRINT AT y+2,x+xo;" ";LET x=x-1: LET I$(y+yo,x+xo+2)=" ": IF
x=0 THEN LET x=0
1046 IF (c/8)+(c/9)=0 THEN LET x=x-(c/8)+(c/9): LET y=y+(x/31)-(x/0): LET x=x-32
*(x/31)+32*(x/0): GO TO 1056
1047 IF c/7 THEN GO TO 1091
1048 IF c/226 THEN LET poc=poc+1: GO SUB 1005: LET poc=poc-1: GO TO 1021
1049 IF NOT ct THEN GO TO 1035
1050 IF c/197 THEN GO TO 1029
1051 IF c/204 THEN GO TO 1010
1052 IF c/195 AND poc=0 AND (y+yo)=(poc+1) THEN FOR p=(y+yo) TO poc+1: LET I$(p
)=I$(p+1): NEXT p: LET poc=poc+1: GO TO 1058
1053 IF c/198 THEN LET yo=yo+18: LET y=1: LET x=0: GO TO 1058
1054 IF c/172 THEN GO TO 1030
1055 IF (c/10)+(c/11)=0 THEN LET y=y+(c/10)-(c/11)
1056 LET yoy=yo: LET yo=yo+18*(y/18)-18*(y/1): LET y=y-18*(y/18)+18*(y/1): IF yo
<1 THEN LET yo=1
1057 IF yoy=yo THEN GO TO 1033
1058 FOR i=1 TO 18: PRINT AT i+2,0;I$(i+yo): NEXT i: GO TO 1032
1059 REM Zmena verze*****
1060 CLS : LET poc=poc+1: FOR i=2 TO poc: LET z$=I$(i, TO 16): LET I$(i, TO 16)=
I$(i,17 TO ): LET I$(i,17 TO )=z$: NEXT i: LET x=0: LET y=1: LET yo=1: LET w$="3
": GO SUB 1005: LET poc=poc-1: GO TO 1058
1061 REM LOAD/SAVE*****
1062 CLS : BEEP .1,30: PRINT m$;AT 9,8;TAB 24;AT 10,8;" LOAD/SAVE(1/s) ";AT 11,8
;TAB 24: PAUSE 0: LET c=PEEK 23560: LET w$=CHR$ c: GO TO 1062+9*(w$="1")+13*(w$=
"s")
1063 INPUT "Název:"; LINE Z$: IF Z$="" THEN BEEP .2,1: GO TO 1063
1064 IF LEN Z$>10 THEN LET Z$=(Z$ TO 10))
1065 RETURN
1066 BEEP .1,35: PRINT u$;AT 9,4;TAB 28;AT 10,4;" Kazet.(0)-M/drive(1-8) ";AT 11
,4;TAB 28: PAUSE NOT P1: LET c=PEEK 23560: LET w$=CHR$ c: IF w$="0" OR w$="8" TH
EN BEEP .2,1: GO TO 1066
1067 LET d=VAL w$
1068 RETURN
1069 PRINT "Verify (a/n)"; PAUSE 0: IF INKEY$="A" OR INKEY$="a" THEN PRINT AT 1
9,0;" Pretoc kazetu na zacatek, pote PLAY": VERIFY Z$ DATA I$(0)
1070 GO TO 1021
1071 GO SUB 1066: IF NOT d THEN GO TO 1073
1072 PRINT AT 0,0: CAT 1: GO SUB 1063: LOAD "m";d;z$ DATA I$(0): GO TO 1074

```

```

1073 GO SUB 1063: LOAD Z$ DATA I$(0)
1074 LET poc=VAL I$(1): GO TO 1021
1075 LET I$(1)=STR$ poc
1076 GO SUB 1066: IF NOT d THEN GO TO 1078
1077 PRINT AT 0,0: CAT 1: GO SUB 1063: SAVE "m";d;z$ DATA I$(0): PRINT AT 12,4:
Verifv...: VERIFY "m";d;z$ DATA I$(0): GO TO 1021
1078 GO SUB 1063: SAVE Z$ DATA I$(0): GO TO 1069
1079 REM Vytisk pres ZX Interface 1
1080 CLS : POKE 65523,29: RANDOMIZE USR 65500: PRINT u$;AT 6,3;TAB 29;AT 7,3: " U
POZORNENI":TAB 29;AT 8,3;TAB 29;AT 9,3: " Program je upraven pro":TAB 29;AT 10,3
: " tisk pres ZX Interface 1":TAB 29;AT 11,3;TAB 29;AT 21,0;TAB 3: "Zmackni-Jake
koliv tlačitko":TAB 32: PAUSE 0
1090 FORMAT "t":300: OPEN #4:"t": FOR p=2 TO poc: PRINT #4:I$(p): NEXT p: CLOSE
#4: CLS : GO TO 1021
1091 REM Instrukce*****
1092 CLS : PRINT AT 4,11:"INSTRUKCE": PRINT : PRINT "Nize uvedene operace je moz
no volat primo ze SLOVNIKU. "
1093 PRINT AT 10,0:"OR'.....zpět na první stránku""AND'.....na další st
ránku""AT'.....na poslední stránku""STOP'.....menu""NOT
'.....vymazání řádku""TO'.....hledání slova"
1094 PRINT u$;AT 21,3: " ENTER=navrát ke slovníku "
1095 PAUSE 0: LET c=PEEK 23560: IF c=13 THEN CLS : GO TO 1058
1096 GO TO 1095
1097 REM Zkousení slov*****
1098 CLS : PRINT u$;AT 10,0: " Slovo:","Ekvivalent:":TAB 32: LET n=0: LET a=1: LE
T o=-1: LET k=0: LET ok=n: GO TO 1113
1099 RANDOMIZE : LET z=INT (RND*poc)+2: LET y=INT (RND*2)
1100 IF y THEN LET xo=17: LET ox=1: GO TO 1102
1101 LET xo=1: LET ox=17
1102 PRINT u$;AT 11,0: " :I$(z,xo TO xo+15):TAB 32: INPUT "Ekvivalent:": LINE z$
: IF z$="" OR z$=" " THEN BEEP :2,20: GO TO 1102
1103 PRINT u$;AT 11,16;z$:TAB 32: LET x=0
1104 FOR p=1 TO 16: LET q=CODE I$(z,ox+p TO ox+p): IF q=32 OR q=44 OR q=59 THEN
LET sd=p: GO TO 1106
1105 NEXT p
1106 LET c=0
1107 IF z$=I$(z,ox TO ox+sd-1) THEN LET ok=a: GO TO 1110
1108 IF NOT c THEN LET ox=ox+sd+1: LET sd=LEN z$: LET c=1: GO TO 1107
1109 LET ok=n
1110 IF NOT ok THEN PRINT m$;AT 2,11;TAB 20;AT 3,11: " SPATNE! ":AT 4,11;TAB 20:
POKE 65523,28
1111 IF ok THEN PRINT INK 9: PAPER 3: FLASH 1:AT 2,11;TAB 20;AT 3,11: " SPRAVNE "
:AT 4,11;TAB 20: POKE 65523,29
1112 PRINT AT 17,0:"Spravne:"I$(z): RANDOMIZE USR 65500
1113 LET o=o+1: LET k=k+1 AND ok: PRINT u$;AT 8,0: " Pokusu: ":o: " Spravnych: "
:k:TAB 32: PRINT FLASH 1;u$;AT 21,10: " ZKOUSET ? "
1114 PAUSE 0: LET c=PEEK 23560: IF c=226 THEN GO TO 1021
1115 PRINT u$;AT 21,0: " :TAB 32: GO TO 1099
1116 REM BEEP
1117 DATA 017,100,098,038,010,058,072,092,031,031,031,014,254,238,016,237,121,06
7,016,254,037,032,244,028,021,032,232,201,058,072,092

```

II. Pomocné tlačítko EDIT

Z módu 3 se lze prostřednictvím tlačítka EDIT „dovolat pomoci“ v podobě souboru instrukcí, které dále zjednodušují a urychlují jak používání slovníku, tak i jeho úpravy.

Provedení jednotlivých úkonů umožňuje současné stisknutí SYMBOL SHIFT a některého z následujících tlačítek:

OR — zpětné vyvolání 1. stránky z kteréhokoli místa ve slovníku. Použitelné v módu 3 — „čtení slovníku“. Kursor spočívá vlevo nahoře.

AND — vyvolání bezprostředně následující stránky. Použitelné v módu 3 — „čtení slovníku“.

AT — přesun na poslední stránku. Kursor spočívá na začátku posledního slova.

STOP — návrat k základnímu výběru (menu). Použitelné v módech 1,3,6.

NOT — okamžité vymazání celého řádku. Použitelné v módu 3.

TO — vyhledávání určitého slova.

Pozor: Před každým použitím SYMBOL SHIFT s dotyčným tlačítkem je nutno nejdříve vrátit se přes ENTER ke slovníku, a to k módu 3 — čtení slovníku.

KRESLIČ+

Jaroslav Huba

Sandrik 320
96661 Hodruše-Hámre

Tento program umožňuje kresbu elektronických schémat, strojnických výkresů, titulních obrázků k různým programům a podobně, přímo pomocí tlačítek klávesnice počítače. Dokáže automaticky kreslit viaceru druhov čiar, a to buď módem PLOT (bod po bode), alebo módem DRAW (čiarka za čiarkou) — viď význam jednotlivých tlačítek.

Uživatel programu má k dispozici kreslicu plochu o rozměrech čtyřech obrazovek, jedna obrazovka = jeden KVADRANT. Prechody z kvadrantu do kvadrantu sú riadené kursorom programu pohybovom kurzora.

Ďalej je možné kedykoľvek si nadefinovať obrázce (ŠABLONKY), o maximálnej ploche 590 znakov, ktoré sú pri vkladaní do pamäte automaticky číslované a hlavné údaje o nich sú zapísané v MENU ŠABLON, ktoré je užívateľovi k dispozícii kedykoľvek, po stlačení vyvolávacej klávesy. Tento súbor maximálne 10 šablon je možné nahráť na mgf pásku a naopak, je možné si do programu dohráť už hotový šablonový súbor z pásky. Ďalej môžeme nakreslený obrázok nahráť a uchovať na mgf pásku, alebo si môžeme z pásky prehrať obrázok do pamäte počítača. Pri nesprávnej manipulácii vypisuje chybové hlásenia, na BREAK reaguje obvyklým spôsobom. Užívateľ

má taktiež k dispozícii pevne uložený NÁVOD NA POUŽIVANIE, ktorý si môže vyvolať kedykoľvek ho potrebuje. Program okrem BASICu obsahuje aj 4 podprogramy (RUTINY) v strojovom kóde.

UPOZORNENIE: Pred nahratím programu si prepnite počítač do módu malých písmen!

Užívateľské parametre:

Kresliaca plocha: 348 x 508 bodov
Ovládanie: cez tastatúru pomocou 36 tlačítek (malé i veľké znaky)
Počet funkcií: 46
Počet uchovávaných obrazoviek (6079 bajtov): 4
Kapacita RAM pre šablony: 4720 bajtov
Volne miesto pre BASIC: približne 200 bajtov
Volne miesto pre CODE: približne 84 bajtov

Typy kreslených čiar:

Počet farieb: 8

Druh farieb: ink

Ovládanie zmien kvadrantov: automatické podľa polohy kurzora

Indikácia polohy kurzora: x — od 1 do

254

y — od 1 do

174

Technické parametre:

RAMTOP: 35800

Dĺžka programu v strojovom kóde:

29730 bajtov

Počet rutin: 4

Organizácia RAM:

Začiatok	Koniec	Názov	Typ	Dĺžka
23756	35302	SUPERDRAW	LINE 27	11546
35800	41879	I.KVADRANT	CODE	6079
41880	47958	II.KVADRANT	CODE	6079
47959	54037	III.KVADRANT	CODE	6079
54038	58758	ŠABLONKY	CODE	4720
58759	59135	INI58759	CODE	376
59136	59293	PAINT	CODE	158
59323	59348	LASER	CODE	26
59349	59370	rezerva		22
59371	59398	MEMORY+	CODE	28
59399	65478	IV.KVADRANT	CODE	6079
				41192

```

0>CLEAR 35799:LOAD ""SCREEN$
:BEEP .6,20:BEEP .3,40:FOR x=1TO
4:BEEP .7,30:LOAD ""CODE :NEXT
x:BEEP .5,35:LOAD ""
27 LET PR=VAL "0": DIM T(VAL "
4"): LET UA=VAL "54038": LET T(1
)=VAL "35800": CLS : PRINT BRIG
HT 1; FLASH 1;AT VAL "6",VAL "6"
;"K R E S L I C + + +"; FLASH 0;
BRIGHT 0: LET KLP=1: LET ODK=VA
L "16384": LET DLZ=VAL "6079": P
OKE VAL "59324",VAL "10": RANDOM
IZE USR VAL "59323": LET CQR=1:
PRINT BRIGHT 1; FLASH 1;AT VAL
"10",VAL "10": "S U P E R "; FLAS
H 0; BRIGHT 0: PAUSE VAL "100":
CLS
28 PRINT AT VAL "20",0;"NEVYPI
NAJ MAGNETOFON": LOAD ""SCREEN$
: PRINT FLASH 1;"STLAC KLAVES";
FLASH 0: PAUSE 0
29 LET ZRAD=0: GO SUB VAL "700
": LET T(2)=VAL "41880": LET T(3
)=VAL "47959": LET T(4)=VAL "593
99"
30 PAPER VAL "7": LET HB=VAL
"0": LET HR=VAL "0": CLS : BORDE
R VAL "5": LET x=VAL "127": LET
y=VAL "87": LET zw=VAL "1": LET
r=VAL "0": LET tocka=VAL "1": LE
T bip=0: LET ink=0: LET ax=VAL "
127": LET ay=VAL "87"
31 DIM H(VAL "1",VAL "1"): DIM
A(VAL "11",VAL "1"): DIM B(VAL
"11",VAL "1"): DIM C(VAL "12",VA
L "1"): DIM X(VAL "11"): DIM Y(V
AL "11"): DIM U(VAL "12"): DIM b
$(VAL "11",VAL "6")
32 INPUT "Medzery:";zrr
33 PRINT #1;"Stlac "; BRIGHT 1
;"ERTDFXCVCUIOPGLJZNMBS 0-7"; BRI
GHT 0
34 PRINT #1;AT 0,0;"x=";x;TAB
VAL "10";"y=";y;TAB VAL "20";"QD
R=";CQR
35 LET s$=INKEY$: IF s$="" THE
N GO TO VAL "35"
36 LET zr=1: IF s$="E" OR s$="
R" OR s$="T" OR s$="D" OR s$="F"
OR s$="X" OR s$="C" OR s$="V" T
HEN LET zr=zrr
38 IF s$="a" THEN GO TO VAL "
500"
39 IF s$="0" THEN LET KLP=1
40 IF s$="r" OR s$="R" THEN L
ET dx=0: LET dy=1*zr: GO TO VAL
"200"
41 IF s$="G" THEN LET KLP=2
42 IF s$="t" OR s$="T" THEN L
ET dx=1*zr: LET dy=1*zr: GO TO V
AL "200"
45 IF s$="f" OR s$="F" THEN L
ET dx=1*zr: LET dy=0: GO TO VAL
"200"
47 IF s$="v" OR s$="V" THEN L
ET dx=1*zr: LET dy=-1*zr: GO TO
VAL "200"
50 IF s$="c" OR s$="C" THEN L
ET dx=0: LET dy=-1*zr: GO TO VAL
"200"
52 IF s$="x" OR s$="X" THEN L
ET dx=-1*zr: LET dy=-1*zr: GO TO
VAL "200"
54 IF s$="d" OR s$="D" THEN L
ET dx=-1*zr: LET dy=0: GO TO VAL
"200"
56 IF s$="e" OR s$="E" THEN L
ET dx=-1*zr: LET dy=1*zr: GO TO
VAL "200"

```

```

58 IF s$="u" THEN LET zw=0: L
ET r=0: GO TO VAL "33"
60 IF s$="i" THEN LET zw=1: L
ET r=0: GO TO VAL "33"
62 IF s$="j" THEN LET zw=0: L
ET r=1: GO TO VAL "33"
64 IF s$="g" THEN INPUT "meno
?";l$: SAVE l$SCREEN$: GO TO VA
L "33"
65 IF s$="l" THEN INPUT AT 0,
0;"Load meno?";l$: LOAD l$SCREEN
$: GO TO VAL "33"
66 IF s$="p" THEN GO TO VAL "
30"
68 IF s$="o" THEN LET bip=NOT
bip: GO TO VAL "33"
70 IF s$>="0" AND s$<="7" THEN
LET ink=VAL s$: GO TO VAL "33"
72 IF s$="n" THEN LET xs=x: L
ET ys=y: GO TO VAL "33"
75 IF s$="b" THEN INPUT #1;AT
0,0;TAB VAL "20";"Stupne: ";so:
PLOT xs,ys: DRAW INK ink; OVER
r;x-xs,y-ys,so/VAL "180"PI
80 IF s$<>"m" THEN GO TO VAL
"140"
90 LET rad=SQR ((x-xs)*(x-xs)+
(y-ys)*(y-ys)): IF rad=0 THEN G
O TO VAL "33"
95 IF xs+rad>VAL "254" THEN G
O TO VAL "33"
97 IF xs-rad<1 THEN GO TO VAL
"33"
100 IF ys+rad>VAL "174" THEN G
O TO VAL "33"
110 IF ys-rad<1 THEN GO TO VAL
"33"
120 CIRCLE INK ink; OVER r;xs,
ys,rad
140 IF s$="z" THEN GO SUB VAL
"9995": GO TO VAL "33"
141 IF s$="s" THEN GO TO VAL "
32"
142 IF s$="K" OR s$="k" THEN G
O TO VAL "540"
143 IF s$="h" OR s$="H" THEN G
O TO VAL "600"
144 IF s$="w" OR s$="W" THEN G
O SUB VAL "700"
145 IF s$="q" OR s$="Q" THEN G
O TO VAL "800"
146 IF s$="Z" THEN GO TO VAL "
8040"
147 IF s$="B" THEN CLS
148 GO TO VAL "33"
200 IF x+dx>=VAL "255" THEN GO
TO VAL "8320"
205 IF x+dx<=0 THEN GO TO VAL
"8350"
210 IF y+dy>=VAL "175" THEN GO
TO VAL "8370"
215 IF y+dy<=0 THEN GO TO VAL
"8400"
240 IF r THEN PLOT OVER 1;x,y
250 IF zw AND NOT tocka THEN P
LOT OVER 1;x,y
260 LET x=x+dx: LET y=y+dy
270 LET tocka=POINT (x,y)
300 IF KLP=1 THEN PLOT INK in
k;x,y
301 IF KLP=2 THEN GO TO VAL "8
020"
302 IF KLP=2 THEN DRAW INK in
k;dx,dy
310 IF bip THEN BEEP VAL ".012
",VAL "44"
320 GO TO VAL "33"
500 IF INKEY$<>"" THEN GO TO V
AL "500"
501 IF INKEY$="" THEN GO TO VA
L "501"
510 PRINT INK ink; OVER r;AT A
BS ((174-y)/8-.45)*x/8;INKEY$
515 IF x<247 THEN LET x=x+8

```

```

520 IF INKEY$<>"" THEN GO TO V
AL "520"
530 GO TO VAL "33"
540 IF PR=1 THEN GO TO VAL "57
0"
544 LET HB=HB+1
545 IF HB>VAL "10" THEN GO TO
VAL "8300"
547 GO SUB VAL "8098": GO SUB V
AL "8100"
550 INPUT "VLOZ ROZMERY SABLONK
Y V ZNAKOCHE,SIRKA=";X(HB);" VYS
KA=";Y(HB)
551 INPUT "VLOZ MENO SABLONKY (
max 6 znakov);b$(HB): LET U(1)=
VAL "54038"
552 LET U(HB+1)=U(HB)+(X(HB)*Y(
HB)*8)
554 IF U(HB+1)>VAL "58758" THEN
GO TO VAL "582"
556 LET ax=x: LET ay=y: LET x=1
: LET y=1: LET PAP=6
557 FOR J=21 TO 22-Y(HB) STEP -
1: FOR I=0 TO X(HB)-1: PRINT PA
PER PAP;AT J,I;" ": NEXT I: NEXT
J: LET PR=1
558 IF ZRAD=1 THEN LET b$(HB)=
"": LET X(HB)=0: LET Y(HB)=0: LE
T HB=HB-1: LET ZRAD=0
560 PRINT #1;AT 0,0;"CISLO SABL
ONKY="; FLASH 1; BRIGHT 1;HB; B
RIGHT 0; FLASH 0: PAUSE 100: BEE
P .5,10
567 LET KR=0: LET PR=1
568 IF PAP=7 THEN GO TO VAL "5
85"
569 GO TO VAL "33"
570 LET X1=1: LET Y1=1: LET U(X
HB)=UA: LET X2=X(HB): LET Y2=Y(H
B): LET PAP=7: LET zw=1: LET r=0
575 GO SUB VAL "900"
580 GO TO VAL "557"
582 LET PAP=7: PRINT #1; FLASH
1;AT 0,0;"POZOR PREKRACUJE KAPAC
ITU RAM!": PAUSE VAL "100": PRIN
T #1;AT 0,0;"
": LET U(HB+1)=0: L
ET ZRAD=1: GO TO VAL "558"
585 GO SUB VAL "8099": GO SUB V
AL "8100": GO TO VAL "8200"
600 LET PR=0: INPUT "CISLO SABL
ONKY?=";K
610 IF (INT (X/8)+X(K))>32 THEN
GO TO VAL "690"
620 IF (INT (Y/8)+Y(K))>22 THEN
GO TO VAL "695"
630: LET X2=INT ((X/8+1)+(X(K)-
1)): LET Y2=INT ((Y/8+1)+(Y(K)-
1)): LET KR=1: LET X1=INT (X/8+1)
: LET Y1=INT (Y/8+1): LET UAX=U(
K): GO SUB VAL "900"
632 LET zw=1: LET r=0: LET PR=0
640 GO TO VAL "33"
690 RANDOMIZE USR VAL "59323":
PRINT #1;AT 0,0;"POSUN KURZOR DO
LAVA 0 ";((X+(X(HB)*8))-VAL "254
"): PAUSE VAL "150": PRINT #1;AT 0
,0;"
": GO T
O VAL "33"
695 RANDOMIZE USR VAL "59323":
PRINT #1;AT 0,0;"POSUN KURZOR DO
LU 0 ";((Y+(Y(HB)*8))-VAL "174")
: PAUSE VAL "150": PRINT #1;AT 0
,0;"
": GO TO VAL "33"
700 GO SUB VAL "8098": GO SUB V
AL "8100": RANDOMIZE USR VAL "59
371"
703 CLS : RANDOMIZE USR VAL "59
323"
705 PRINT AT 0,8;"NAVOD NA POUZ
ITIE";AT 2,4;"NA ZACIATKU SI PRE
PNITE";AT 3,4;"POCITAC DO MODU
MALYCH";AT 4,4;"PISMEN - ";CAPS

```

```

SHIFT";" + ";"2"
706 PRINT AT 8,2;"ERTDFXCV";" - POHYB KURZORU".
710 PRINT AT 10,2;"u";" - KRESL ENIE";AT 10,17;"j";" - GUMOVANIE";AT 12,2;"i";" - POHYB BEZ KRES BY";AT 14,2;"p";" - HORUCI START";AT 16,2;"g";" - SAVE SCR$".
715 PRINT AT 16,21;"z";" - COPY";AT 17,24;"CONSUL";AT 19,2;"1";" - LOAD SCR$";AT 19,19;"s";" - MEDZERY".
720 PRINT #1; FLASH 1;AT 0,0;"S TLAC KLAVES!"; FLASH 0: PAUSE 0
725 RANDOMIZE USR VAL "59323": CLS : PRINT AT 1,2;"n-m";" - KRU ZNICA-STRED = n";AT 2,15;"POLOME R = m".
730 PRINT AT 4,2;"a";" - PISANI E ZNAKOV";AT 6,2;"b";" - STUPNE SPOJENIA BODOV".
735 PRINT AT 8,2;"K-k";" - VKLA DANIE SABLON";AT 10,2;"H-h";" - VYBERANIE Z RAM NA SCR$";AT 12,5;"(PODLA KURZORU...)";AT 14,2;"Q-q";" - VYPIS MENU SABLON";AT 18,2;"W-w";" - NAVOD";AT 16,2;"LOAD/SAVE SABLONY".
750 PRINT #1,AT -1,0;"
": PRINT #1,AT 0,0;"STLAC KLAVESU!"; PAUSE 0: CLS : RANDOMIZE USR VAL "59323".
755 PRINT AT 2,2;"o - BEEP";AT 4,2;"D - PLOT";AT 6,2;"G - DRAW".
760 PRINT AT 8,2;"Z";" - FILL";AT 10,2;"B";" - CLS".
763 PRINT #1,AT 0,0;"STLAC KLAVESU": PAUSE 0: GO SUB VAL "8099": GO SUB VAL "8100".
764 CLS : GO SUB VAL "8099": GO SUB VAL "8100": RANDOMIZE USR VAL "59371".
765 RANDOMIZE USR VAL "59323": RETURN
800 LET PR=0: INPUT "CHCES MENU ? a/n";a$
810 IF a$="a" OR a$="A" THEN GO TO VAL "820"
815 GO TO VAL "33"
820 RANDOMIZE USR VAL "59323": GO SUB VAL "8098": GO SUB VAL "8100": CLS
826 PRINT AT 0,10;"M E N U";AT 1,0;"Stav obsadenia pamate sablonkami";AT 3,1;"CISLO".
830 PRINT AT 3,8;"MENU";AT 3,14;"VYS * SIR";AT 3,25;"ADRESA".
835 FOR u=5 TO HB+5
836 PRINT AT u,3;(u-4);AT u,7;b$(u-4);AT u,15;Y(u-4);AT u,20;X(u-4);AT u,25;U(u-4)
837 NEXT u
840 INPUT "SAVE - LOAD SABLONKY s/1";a$
845 IF a$="s" OR a$="S" THEN GO TO VAL "850"
847 IF a$="1" OR a$="L" THEN GO TO VAL "870"
849 GO SUB VAL "8099": GO SUB VAL "8100": GO TO VAL "34"
852 LET H(1,1)=HB: FOR L=1 TO HB: LET A(L,1)=X(L): LET B(L,1)=Y(L): LET C(L,1)=U(L): NEXT L
853 LET C(HB+1,1)=U(HB+1)
854 INPUT "NAZOV SAVE SUBORU ";a$: SAVE a$ DATA H(): SAVE "X" DATA A(): SAVE "Y" DATA B(): SAVE "U" DATA C(): SAVE "b$" DATA b$(): SAVE a$CODE U(1),U(HB)-U(1)-1: GO TO VAL "849"
870 PRINT #1,AT 0,0;"ZAPNI MAGNETOFON!": LOAD "" DATA H(): LOAD "" DATA A(): LOAD "" DATA B(): LOAD "" DATA C(): LOAD "" DATA b$(): LOAD ""CODE
872 PRINT #1;AT 0,0;"CAKAJ, UKLADAM!".
875 LET HB=H(VAL "1",VAL "1"): FOR L=VAL "1" TO HB: LET X(L)=A(
L,VAL "1"): LET Y(L)=B(L,VAL "1"): LET U(L)=C(L,VAL "1"): NEXT L
880 LET U(HB+1)=C(HB+1,1): LET UA=U(HB+1): GO TO VAL "849"
900 PRINT #1; FLASH 1; BRIGHT 1;AT 0,0;"WAIT PLEASE!"; BRIGHT 0; FLASH 0: DATA 1,6,1,6,VAL "920",7,14,7,14,VAL "960",15,22,15,22,VAL "1000",1,6,7,14,VAL "1010",7,14,15,22,VAL "1050",1,6,15,22,VAL "1100"
905 RESTORE : FOR H=1 TO 6
910 READ H1,H2,H3,H4,CR
912 IF Y1>=H1 AND Y1<=H2 AND Y2>=H3 AND Y2<=H4 THEN GO TO CR
915 NEXT H
920 REM ** zona 1 **
921 LET SH=VAL "22432": LET HR=1: LET Z=1: GO SUB VAL "930": GO SUB VAL "8000": RETURN
930 LET D=X2-X1: LET C=SH-(Y1-HR)*32+256: LET A=Y2-Y1+1: LET E=X1-1
935 REM ** generovanie adries **
936 FOR P=Z TO A: FOR R=1 TO 8: LET PA=C-(R*256)+E
940 IF KR=0 THEN GO SUB VAL "1200": IF KR=0 THEN LET UA=UA+D+1
945 IF KR=1 THEN GO SUB VAL "1300": IF KR=1 THEN LET UAX=UAX+D+1
950 NEXT R: LET C=C-32: NEXT P:
959 RETURN
960 REM ** zona 2 **
961 LET SH=VAL "20448": LET HR=7: LET Z=1: GO SUB VAL "930": GO SUB VAL "8000": RETURN
1000 REM ** zona 3 **
1001 LET SH=VAL "18400": LET HR=15: LET Z=1: GO SUB VAL "930": GO SUB VAL "8000": RETURN
1010 REM ** zona 1 - 2 **
1011 LET Y3=6: LET SH1=VAL "22432": LET HR1=1: LET SH2=VAL "20448"
1015 LET C2=SH2+VAL "256": LET C1=SH1-(Y1-HR1)*VAL "32"+VAL "256": LET D=X2-X1: LET A=Y3-Y1+1: LET F=Y2-Y1+1
1020 LET C=C1: LET Z=1: LET E=X1-1: GO SUB VAL "935": LET C=C2: LET Z=A+1: LET A=F: GO SUB VAL "935": GO SUB VAL "8000": RETURN
1050 REM ** zona 2 - 3 **
1051 LET Y3=14: LET SH1=VAL "20448": LET HR1=7: LET SH2=VAL "18400": GO TO VAL "1015"
1100 REM ** zona 1 - 2 - 3 **
1101 LET C1=(VAL "22432")-(Y1-1)*32+256: LET C2=VAL "20704": LET C3=VAL "18656": LET A=6-Y1+1: LET F=Y2-Y1+1: LET D=X2-X1: LET C=C1: LET Z=1: LET E=X1-1: GO SUB VAL "935": LET C=C2: LET Z=1: LET A=8: GO SUB VAL "935": LET C=C3: LET Z=1: LET A=Y2-14: GO SUB VAL "935": GO SUB VAL "8000": RETURN
1200 BEEP .05,35: LET ODK=PA: LET DLZ=D+1: LET KAM=UA: GO SUB VAL "8100": RETURN
1300 BEEP .05,VAL "40": LET ODK=UAX: LET DLZ=D+1: LET KAM=PA: GO SUB VAL "8100": RETURN
8000 IF KR=0 THEN LET UA=UA-D
8010 IF KR=1 THEN LET UAX=UAX-D
8012 PRINT #1;AT 0,0;"
"
8015 RETURN
8020 IF x+dx>VAL "254" THEN GO TO VAL "8030"
8022 IF x+dx<1 THEN GO TO VAL "8030"
8024 IF y+dy>174 THEN GO TO VAL "8030"
8026 IF y+dy<1 THEN GO TO VAL "8030"
8028 GO TO VAL "302"
8030 LET KLP=1: GO TO VAL "300"
8040 RANDOMIZE USR VAL "59323": INPUT "Farba INK vyplne";FAR
8042 IF FAR<0 OR FAR>7 THEN GO TO VAL "8040"
8044 POKE VAL "59293",FAR+56
8046 RANDOMIZE USR VAL "59136"
8048 GO TO VAL "33"
8055 RETURN
8098 LET ODK=VAL "16384": LET DLZ=VAL "6079": LET KAM=T(CQR): RETURN
8099 LET ODK=T(CQR): LET DLZ=VAL "6079": LET KAM=VAL "16384": RETURN
8100 RANDOMIZE ODK: POKE VAL "59380",PEEK VAL "23670": POKE VAL "59381",PEEK VAL "23671": RANDOMIZE DLZ: POKE VAL "59386",PEEK VAL "23670": POKE VAL "59387",PEEK VAL "23671": RANDOMIZE KAM: POKE VAL "59383",PEEK VAL "23670": POKE VAL "59384",PEEK VAL "23671": RANDOMIZE USR VAL "59371": RETURN
8200 LET PR=0: LET x=ax: LET y=ay: GO TO 33
8300 RANDOMIZE USR VAL "59323"
8305 PRINT #1; BRIGHT 1;AT 0,0;"KONIEC SABLON!"; BRIGHT 0
8310 PAUSE VAL "100"
8315 RANDOMIZE USR VAL "59323": GO TO 33
8320 GO SUB VAL "8098": GO SUB VAL "8100": IF CQR=1 THEN LET CQR=2: LET X=1
8325 IF CQR=3 THEN LET CQR=4: LET X=1
8330 GO SUB VAL "8099": GO SUB VAL "8100": GO TO VAL "300"
8350 GO SUB VAL "8098": GO SUB VAL "8100"
8355 IF CQR=2 THEN LET CQR=1: LET X=254
8360 IF CQR=4 THEN LET CQR=3: LET X=254
8365 GO SUB VAL "8099": GO SUB VAL "8100": GO TO VAL "300"
8370 GO SUB VAL "8098": GO SUB VAL "8100": IF CQR=3 THEN LET CQR=1: LET Y=1
8375 IF CQR=4 THEN LET CQR=2: LET Y=1
8380 GO SUB VAL "8099": GO SUB VAL "8100": GO TO VAL "300"
8400 GO SUB VAL "8098": GO SUB VAL "8100": IF CQR=1 THEN LET CQR=3: LET Y=174
8410 IF CQR=2 THEN LET CQR=4: LET Y=174
8420 GO SUB VAL "8099": GO SUB VAL "8100": GO TO VAL "300"
9990 FOR Q=POZY1/2 TO POZY2/2-2: LPRINT " "; NEXT Q
9991 IF AAA+BBB=2 THEN LPRINT " $"; RETURN
9992 IF AAA=1 THEN LPRINT "o"; RETURN
9993 IF BBB=1 THEN LPRINT "o"; RETURN
9995 FOR W=0 TO VAL "255": LET POZY1=0: FOR G=POZY1 TO VAL "173" STEP 2: LET AAA=POINT (W,G): LET BBB=POINT (W,G+1): IF (AAA+BBB)>0 THEN LET POZY2=G: GO SUB VAL "9990": LET POZY1=POZY2
9996 NEXT G: LPRINT : NEXT W: LPRINT : LPRINT "OK": LPRINT : RETURN
9999 CLS : GO TO VAL "900"

```


Výpisy podprogramů ve strojovém kódu („Kreslící“)

INI58759 - inicializačný program pre interface

```

58759 / 33 159 229 62 0 119 35 119
58767 / 33 197 92 62 183 119 62 230
58775 / 35 119 201 0 0 0 0 0
58783 / 0 11 0 0 0 0 0 0
58791 / 22 25 30 240 29 32 253 21
58799 / 32 248 201 0 0 0 0 0
58807 / 33 160 229 52 71 62 144 211
58815 / 127 219 31 203 71 40 30 120
58823 / 211 63 205 167 229 62 1 211
58831 / 95 205 167 229 62 0 211 95
58839 / 205 167 229 205 167 229 126 254
58847 / 98 204 71 230 201 205 84 31
58855 / 56 215 207 255 0 0 0 0
58863 / 0 62 32 205 183 229 13 32
58871 / 248 0 201 0 0 0 0 0
58879 / 33 159 229 126 198 16 119 254
58887 / 84 210 71 230 33 160 229 70
58895 / 144 56 237 79 205 239 229 201
58903 / 87 33 160 229 70 144 56 15
58911 / 79 205 239 229 33 197 92 62
58919 / 183 119 35 62 230 119 201 213
58927 / 62 13 205 187 229 209 24 233
58935 / 0 0 0 0 0 0 0 0
58943 / 0 0 0 0 0 0 0 0
58951 / 33 159 229 62 0 119 35 119
58959 / 62 13 205 187 229 201 0 0
58967 / 0 0 0 0 0 0 0 0
58975 / 245 14 1 205 239 229 241 214
58983 / 164 33 149 0 79 70 121 254
58991 / 0 40 20 120 254 127 48 2
58999 / 24 1 13 35 121 254 255 32
59007 / 236 14 1 205 239 229 201 120
59015 / 254 127 48 9 229 197 205 183
59023 / 229 193 225 24 230 203 191 229
59031 / 197 205 183 229 193 225 24 218
59039 / 0 0 0 0 0 0 0 0
59047 / 14 1 205 239 229 201 0 0
59055 / 0 0 0 0 0 0 0 0
59063 / 254 5 48 1 201 254 24 56
59071 / 5 254 31 48 1 201 254 127
59079 / 56 8 254 164 48 4 205 167
59087 / 230 201 254 23 32 11 33 197
59095 / 92 62 23 119 35 62 230 119
59103 / 201 254 165 56 4 205 95 230
59111 / 201 254 13 32 4 205 71 230
59119 / 201 254 6 32 4 205 255 229
59127 / 201 205 183 229 201

```

PAINT - program pre vyplňanie obrazcov

```

59135 / 0 58 125 92 95 58 126 92
59143 / 87 58 157 231 50 143 92 75
59151 / 66 66 205 84 231 75 4 120
59159 / 254 174 210 49 231 197 213 205
59167 / 206 34 209 193 197 213 205 213
59175 / 45 209 193 60 61 254 1 194
59183 / 17 231 66 75 205 84 231 75
59191 / 5 120 254 1 218 83 231 197
59199 / 213 205 206 34 209 193 197 213
59207 / 205 213 45 209 193 60 61 254
59215 / 1 194 50 231 201 75 197 213
59223 / 205 229 34 209 193 12 121 254
59231 / 255 210 120 231 197 213 205 206
59239 / 34 209 193 197 213 205 213 45
59247 / 209 193 60 61 254 1 194 85
59255 / 231 75 197 213 205 229 34 209
59263 / 193 13 121 254 1 218 156 231
59271 / 197 213 205 206 34 209 193 197
59279 / 213 205 213 45 209 193 60 61
59287 / 254 1 194 121 231 201

```

LASER - akustický efekt

```

59323 / 6 10 197 33 0 3 17 1
59331 / 0 229 205 181 3 225 17 16
59339 / 0 167 237 82 32 240 193 16
59347 / 233 201

```

MEMORY+ - relokatívny prenos dat v pamäti

```

59371 / 245 197 213 229 221 229 253 229
59379 / 33 216 139 17 0 64 1 191
59387 / 23 237 176 253 225 221 225 225
59395 / 209 193 241 201

```

Význam a funkcie tlačítko:

Program je riadený stláčaním kláves, pričom sa rozlišujú malé aj veľké znaky. Veľké znaky sa zadávajú spolu so stláčaním CAPS SHIFT.

Tlačítko Funkcia

- (a) písanie textu do obrázku, polohu prvého znaku nastavíme kurzorom, stlačíme a a znak, ktorý chceme napísať, pred každým ďalším znakom musíme stlačiť a;
- (b) určuje stupne spojenia dvoch rôznych bodov, prvý bod určí-

me kurzorom a stlačením n, druhý tiež kurzorom a stlačením b. Počítač sa opýta na stupne obluku, ktorým má dané body spojiť. Napr. pre priamku zadáme úhol 0;

- (B) funkcia CLS;
- (C) pohyb kurzoru po bodoch kolmo nadol;
- (C) pohyb kurzoru po navolených skokoch (MEDZERY) kolmo nadol;
- (d) pohyb kurzoru vodorovne vľavo;
- (D) pohyb kurzoru vodorovne vľavo po skokoch;
- (e) pohyb kurzoru 45 stupňov vľavo hore;
- (E) pohyb kurzoru 45 stupňov vľavo hore po skokoch;
- (I) pohyb kurzoru vodorovne vpravo;

- (F) pohyb kurzoru vodorovne vpravo po skokoch;
- (S) SAVE obrazovky ako bežne SAVE SCREENS;
- (G) spôsob kreslenia DRAW, pri kreslení kurzor pri stlačení CAPS SHIFT a navolených medzerách kreslí spojitú čiaru po skokoch pri stlačení u, pri stlačení i kreslí čiarkovanú čiaru s prerušovaním o jeden bod (— —);
- (h/(H)) vyvolávanie šablonok z pamäte. Kurzorom sa dostaneme do miesta, kde má byť ľavý dolný roh šablony, stlačíme h (H) a udáme číslo šablony, ktorá sa má nakresliť (vyvolať z RAM);
- (i) pohyb kurzoru bez kreslenia pri móde PLOT, (stlačene O), prerušovaná čiaru pri móde DRAW (stlačene G);
- (j) mazanie starej kresby, bod po bode, prepnete na O;
- (k/(K)) dvojfázove vkládanie šablonok, pri prvom stlačení zadáme výšku a šírku vkládanej šablony a jej názov, potom sa program vráti do režimu kreslenia a šablónku môžeme nakresliť

v ľavom dolnom rohu obrazovky do priestoru, ktorý sa vyfarbí zmenou PAPER. Druhým stlačením k (K) šablonku vložíme do pamäti. Šablonky sú číslované automaticky, maximálny počet je 10. Program je zabezpečený proti vloženiu nesprávnych rozmerov a rozmerov prekračujúcich kapacitu pamäte, ako aj proti prekročeniu počtu šablon;

LOAD obrázku ako bežne **LOAD SCREEN\$**; kresba kružnice, kurzor nastavíme do požadovaného stredu kružnice, potom stlačíme n, kurzor posunieme do miesta, kadiaľ má kružnica prechádzať a stlačíme m; udanie polohy stredu kružnice, udanie prvého bodu pri spájaní bodov pomocou b; pri pohybe kurzoru bude reproduktor piskať; spôsob kreslenia **PLOT**, bod za bodom; znovu nastartovanie programu, **WARM START**;

pohyb kurzoru kolmo nahoru; pohyb kurzoru kolmo nahor po skokoch; udanie medzery (skoku) kurzoru pri súčasnom stlačení **CAPS SHIFT**, dĺžka skoku pri /O/, dĺžka čiar pri /G/; pohyb kurzoru 45 stupňov vpravo hore; pohyb kurzoru 45 stupňov vpravo hore po skokoch;

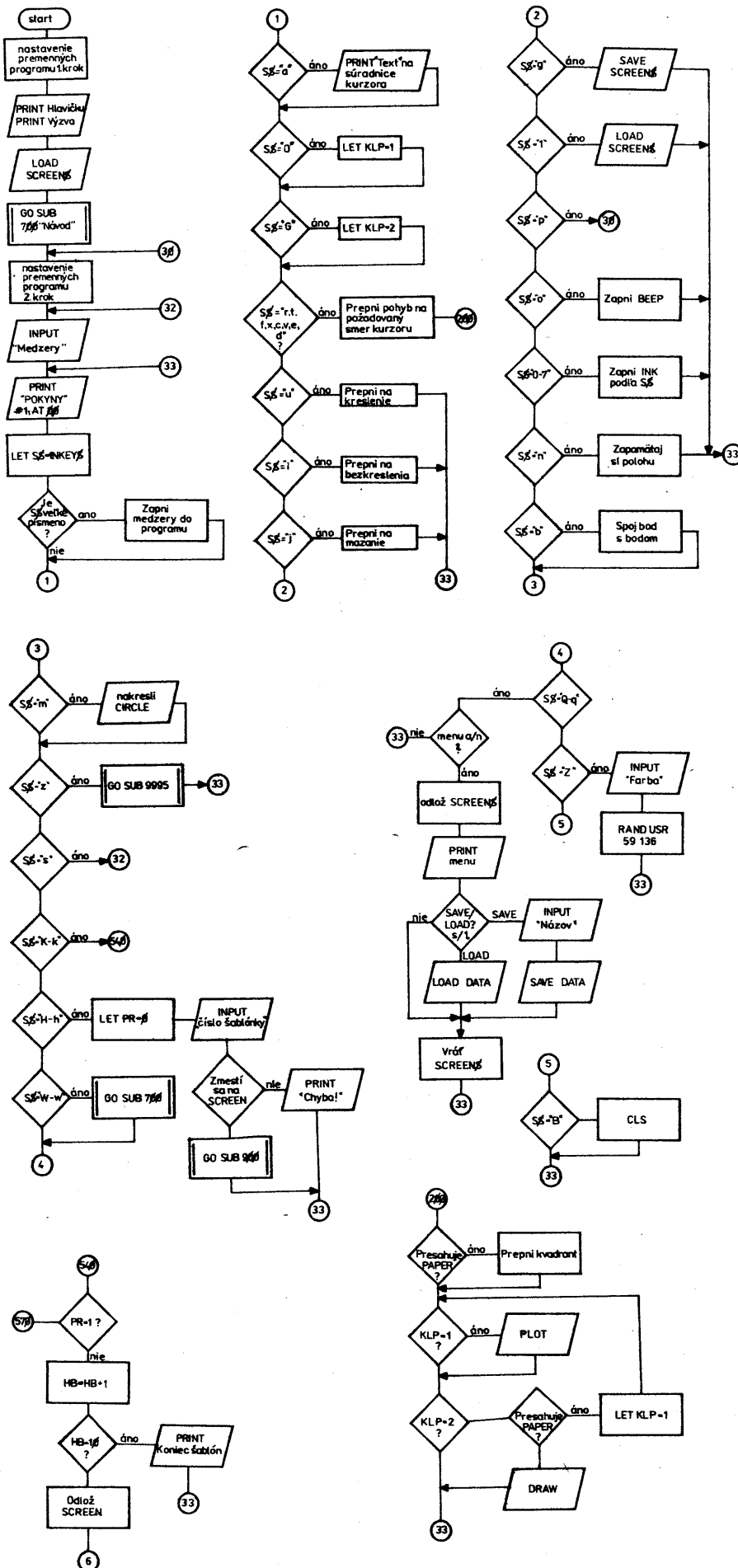
kreslenie podľa kurzora; pohyb kurzora 45 stupňov vpravo dolu; pohyb kurzora 45 stupňov vpravo dolu po skokoch; po stlačení sa inicializuje ovládaci strojový podprogram a obrázok sa skopíruje cez mechanickú strojovú tlačiareň Consul 258;

výplň uzavretých figúr. Nakreslíme nejakú uzavretú krivku (kružnica, štvorec...), vojdeme kurzorom dovnútra a stlačíme Z, potom udáme číslom farbu, ktorou chceme útvár vyplniť;

po stlačení sa objaví na obrazovke výpis **MENU ŠABLON**, údaje o stave doposiaľ vložených šablon. Možno volať kedykoľvek; po stlačení sa na obrazovke objaví výpis **NAVOD NA POUŽIVANIE**. Možno volať kedykoľvek;

pohyb kurzoru 45 stupňov vľavo dolu; pohyb kurzoru 45 stupňov vľavo dolu po skokoch;

stlačením jedného z týchto tlačítkov sa navoli automaticky farba **INK**;



CAPS SHIFT spolu s tlačítkami pre posuv kurzoru umožňuje skákať o dopredu navolené medzery (s).

Užívateľ má k dispozícii plochu štyroch obrazoviek, ktoré sa mu automaticky prepínajú ako náhle sa kurzor priblíži k okraju PAPER. Všetky štyri kvadranty sa ukladajú v RAM a je možné ich jednu po druhej kopírovať (SAVE — g). Vľavo dole, v edičnej zóne, sú neustále zobrazované údaje o okamžitej polohe kurzoru v súradniciach x a y, vpravo zase údaj o momentálnej polohe v kvadrante (QDR= ..). Všetky dôležité stavy sú akusticky indikované a chybové hlásenie sa vypisuje taktiež v edičnej zóne.

Poznámka: Pri prechode kurzorom s kvadrantu do kvadrantu je zakázané a nemožné prejsť cez rohy PAPER, treba prechádzať cez hrany PAPER!

Chybové hlásenia:

KONIEC ŠABLONI — Počet šablon dosiahol 10.

POZOR PREKRAČUJE KAPACITU RAM! — zadaná šablona sa nevnesie do priestoru určeného pre šablony;

POSUN KURZOR DOLU O... — šablónka by sa pri špatnom vyberaní nevnesla do obrazovky, prekročila by horný okraj PAPER;

POSUN KURZOR DOLAVA O... — šablónka prekračuje pravý okraj PAPER.

VYUŽÍVANIE STROJOVÝCH PODPROGRAMOV:

Okrem programu v BASIC využíva KRESLIČ+ aj 4 strojové podprogramy, a to:

INI58759 — inicializačný program pre interfejs,

PAINT — program pre vyplňanie obrazcov,

LASER — akustický efekt,

MEMORY+ — relokatívny prenos dát v pamäti.

Prvá pomoc pri „nabúrání“ programu:

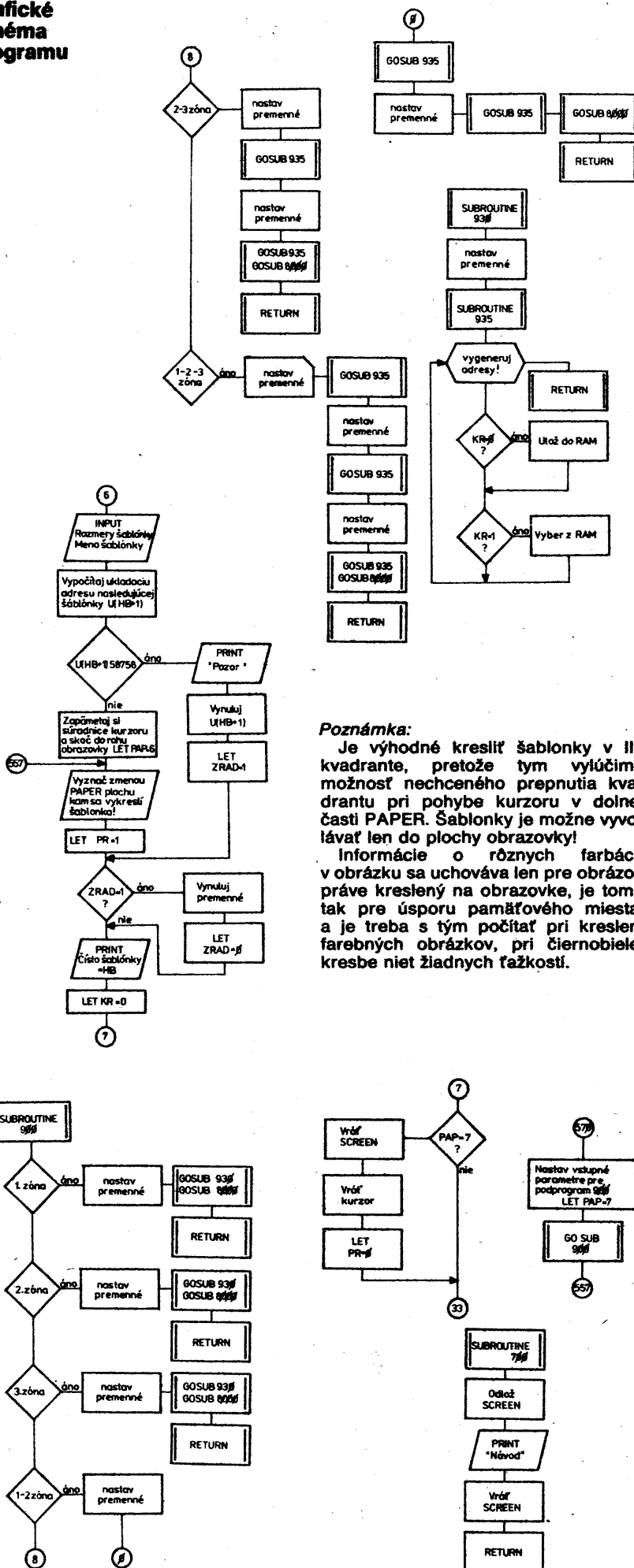
Pokiaľ by pri kreslení došlo k vypádnutiu z programu a CONTINUE by bolo neučinné, znovunabehnutie programu bez akéhokoľvek poškodenia nakresleného obrázku vykonáme odoslaním „nulového“ príkazu GOTO 35.

Pokiaľ systém odmieta prijať informácie cez INPUT, jediná pomoc je RESET.

Práca so súborom šablón:

Keď si chceme vytvorené šablony nahráť na mgf, vyvoláme si cez q (Q) MENU a na dotaz „SAVE—LOAD ŠABLONY? s/I“ odpovíme „s“ a vložíme meno, pod ktorým chceme súbor šablón uložiť na pásku. Potom na obvyklé „Start tape, then press any key...“ stlačíme ľubovoľnú klávesu. LOAD sa prevedie podobne. V menu sa vypisujú aj čísla adries jednotlivých šablón a ich dĺžka, takže je možné program prerušiť BREAK a nahrávať aj jednotlivé šablony, čo je nepraktické a navyše hrozí možnosť ľahkého omylu.

Grafické schéma programu



PROGRAMÁTOR PAMĚTI PGM-2

J. Bezdíček

Mikropočítačové systémy se uplatňují prakticky ve všech oborech a aplikace přibývají. Stoupají požadavky na programové vybavení. Přes množství u nás vyvinutých a předváděných vývojových systémů se však stále setkáváme s jejich nedostatkem. Proto se používají dostupné mikropočítače (mikroprocesor pracuje stejně výkonně ve vývojovém systému jako v „jednoduchém“ mikropočítači; záleží jen na programovém vybavení a pro efektivnost práce ještě na perifériích), na nich se programy odlaďují a odladěné programy často ukládají do paměti ROM. Vznikají různá programovací zařízení, většinou podle typu právě používané paměti. Proto byl navržen a realizován programátor PGM-2, který je řešen modulárně a umožňuje programovat nejrůznější typy paměti. Obecně je možné použít pro řízení PGM-2 libovolný mikropočítač, který má dostačující zónu paměti pro čtená/programovaná data a řídicí program. V článku je popsáno připojení k SAPI 1.

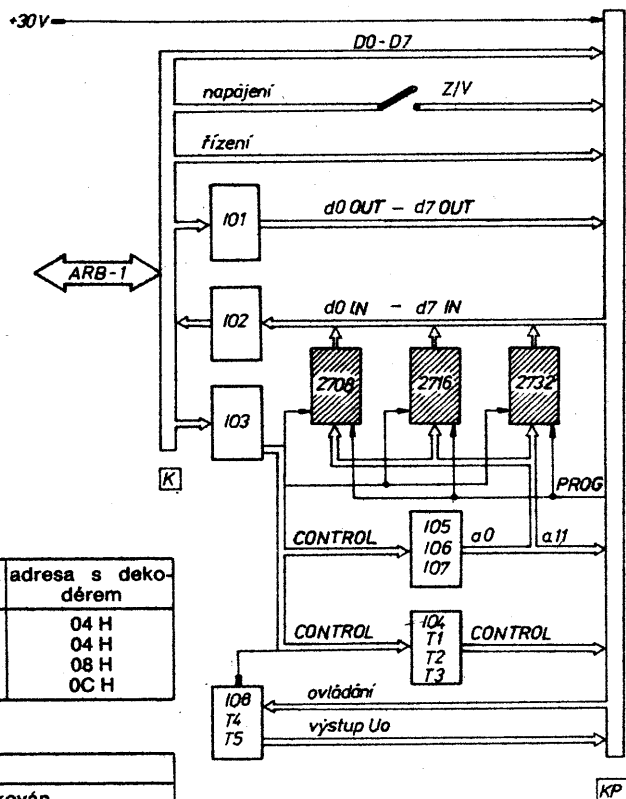
Popis zapojení

PGM-2 se připojuje k SAPI 1 na sběrnici ARB-1 přes konektor K a plochý kabel. Vyžaduje dále připojení ke zdroji stejnosměrného napětí 30 až 70 V, schopnému dodat alespoň 200 mA. Protože i SAPI 1 „spěje“ k paměti RAM 64 kB, byl programátor řešen jako periférie IN/OUT. Signály A5, A6 a A7 přímo řídí výběrové vstupy obvodů styku, společně se signály IOW a IOR. Dodatečně byl programátor doplněn jednoduchým dekodérem (74188), který je umístěn přímo v krytu konektoru K.

K připojení je použito obvodů 3212. Připojit programátor přes programovatelný PPI 8255 se ukázalo nevýhodné. Většinu signálů je stejně zapotřebí zesílit a navíc při přeprogramování 8255 do jiného režimu se výstupy všech portů nastaví na úroveň L (jako po RESET), což by působilo potíže při procesu programování/verifikace v každé adrese.

Na konektoru KP-1 jsou vyvedeny signály a úrovně napětí, potřebné pro všechny režimy práce s pamětmi. Přifazení určuje Tab. 1. Při programování 8748 je nutné rozšířit počet ovládacích

Obr. 1. Blokové zapojení programátoru PGM-2



Tab. 2a. Ovládání obvodů pro styk mikropočítače s PGM-2.

obvod	funkce	instr.	adresa	adresa s dekodérem
IO 1	data pro programování	OUT	20 H	04 H
IO 2	načtená data	IN	20 H	04 H
IO 3	řízení I	OUT	40 H	08 H
IO101	řízení II	OUT	80 H	0C H

Řízení I

bit	IO3	funkce	úroveň	význam
b7	Q8	ovládání stabilizátoru	L	stabil. blokován
b6	Q7	ovládání režimu IO 1	L	velká imped. IO 1
b5	Q6	nulování čítače adr.	L	nulování
b4	Q5	vstup čítače adr.		další vyšší adr.
b3	Q4	ovládání výstupu Q4		Q4
b2	Q3	ovládání tranzistoru T1	H	tranzistor sepnut
b1	Q2	ovládání tranzistoru T2	H	tranzistor sepnut
b0	Q1	ovládání tranzistoru T3	H	tranzistor sepnut

Řízení II

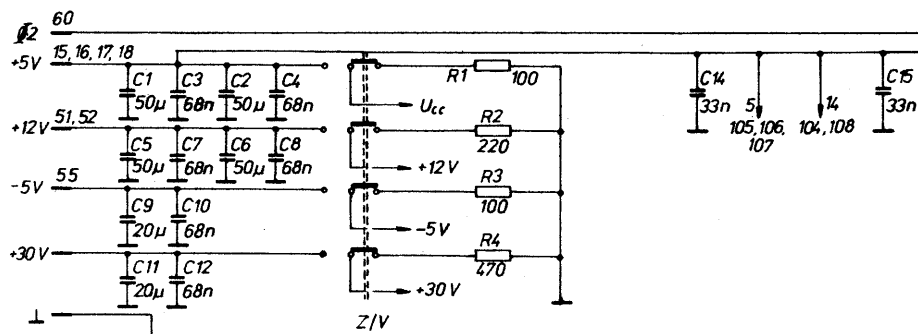
bit	IO101	funkce	úroveň	význam
b7	Q8	výběr obvodu IO102/S2/	H	obvod aktivní
b6	Q7	řízení funkce IO102/S2/	L	velká impedance
b5	Q6	X		
b4	Q5	X		
b3	Q4	X		
b2	Q3	ovládání T0	H	mód programování
b1	Q2	ovládání RESET		latch adresy
b0	Q1	ovládání tranzistoru T101	H	tranzistor sepnut

Tab. 1. Zapojení konektoru PK-1

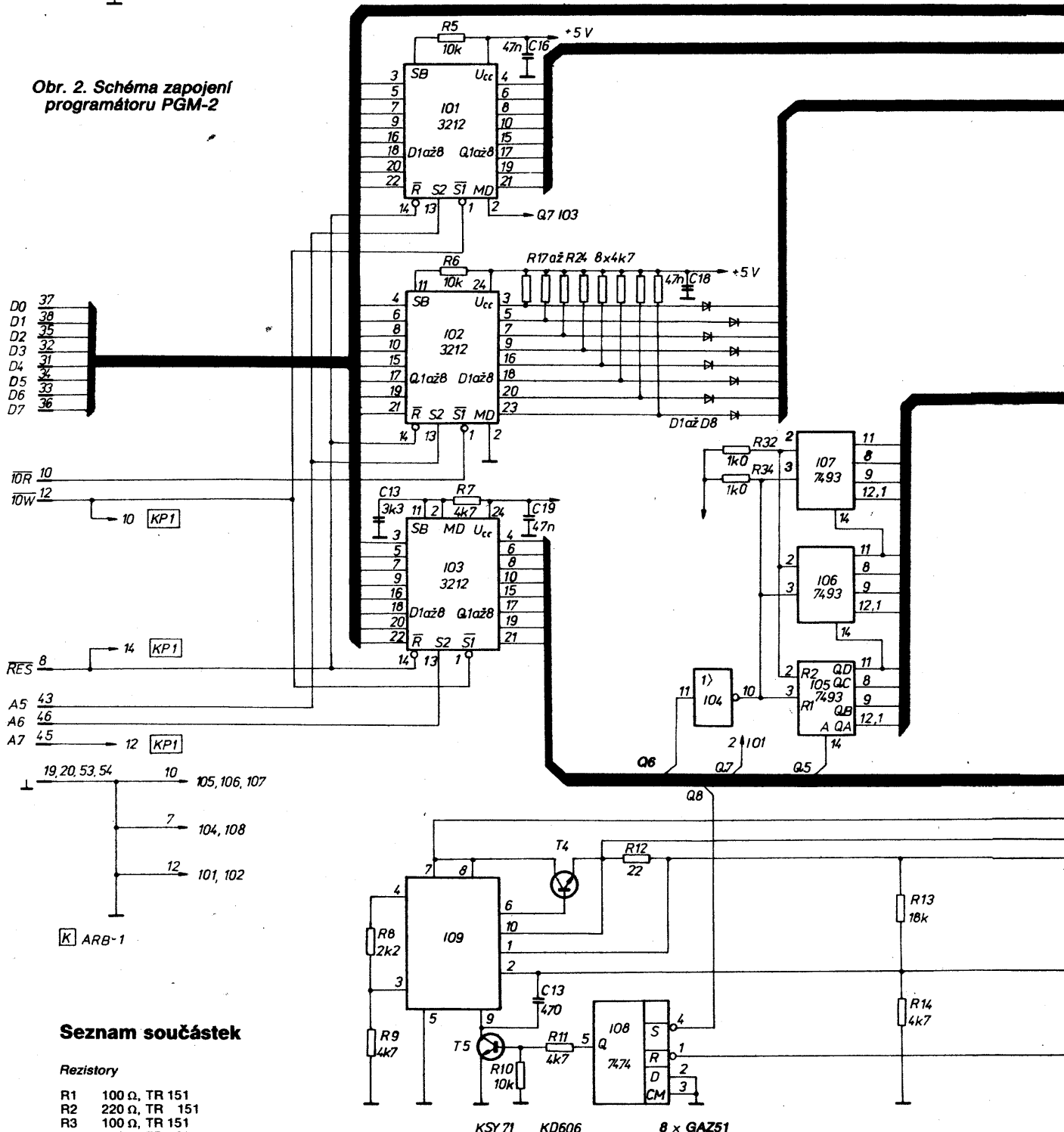
č.	Signál	č.	Signál
1	a 0	2	L
3	a 1	4	+5 V napájení
5	a 2	6	+5 V
7	a 3	8	In nastavení I
9	a 4	10	IOW
11	a 5	12	A 7
13	a 6	14	RES
15	a 7	16	d 7 OUT
17	a 8	18	d 6 OUT
19	a 9	20	d 5 OUT
21	a 10	22	d 4 OUT
23	a 11	24	d 3 OUT
25	Cm čas. konstanta	26	d 2 OUT
27	-5 V	28	d 1 OUT
29	+12 V napájení	30	d 0 OUT
31	Ucc	32	d 7
33	XTAL 1	34	D 6
35	Y výstup 6 IO4	36	D 5
37	A vstup 5 IO4	38	D 4
39	ACE uvolnění U ₀	40	D 3
41	Rn nastavení U ₀	42	D 2
43	U ₀ výstup stabil.	44	D 1
45	KT4 vstup stabil.	46	D 0
47	+30 V napájení	48	d 7 IN
49	Q ₄	50	d 6 IN
51	KT3	52	d 5 IN
53	T2	54	d 4 IN
55	KT2	56	d 3 IN
57	T1	58	d 2 IN
59	KT1	60	d 1 IN
61	L zem	62	d 0 IN

signálů. Proto je na KP-1 vyvedena i datová sběrnice ARB-1 (D0 až D7) a adresový signál A7. Odpojení napájecího napětí pro programované obvody spínačem Z/V umožní jejich bezpečné zasunutí do a vyjmutí z objímky.

Stabilizátor napětí je ovládán z KP-1, a to jak úroveň a proudové omezení, tak uvolnění (vstup R IO8). Blokování se využívá například u 8748 tak, že signálem ALE nepřímo kontrolujeme správnost zasunutí 8748 do objímky



Obr. 2. Schéma zapojení programátoru PGM-2



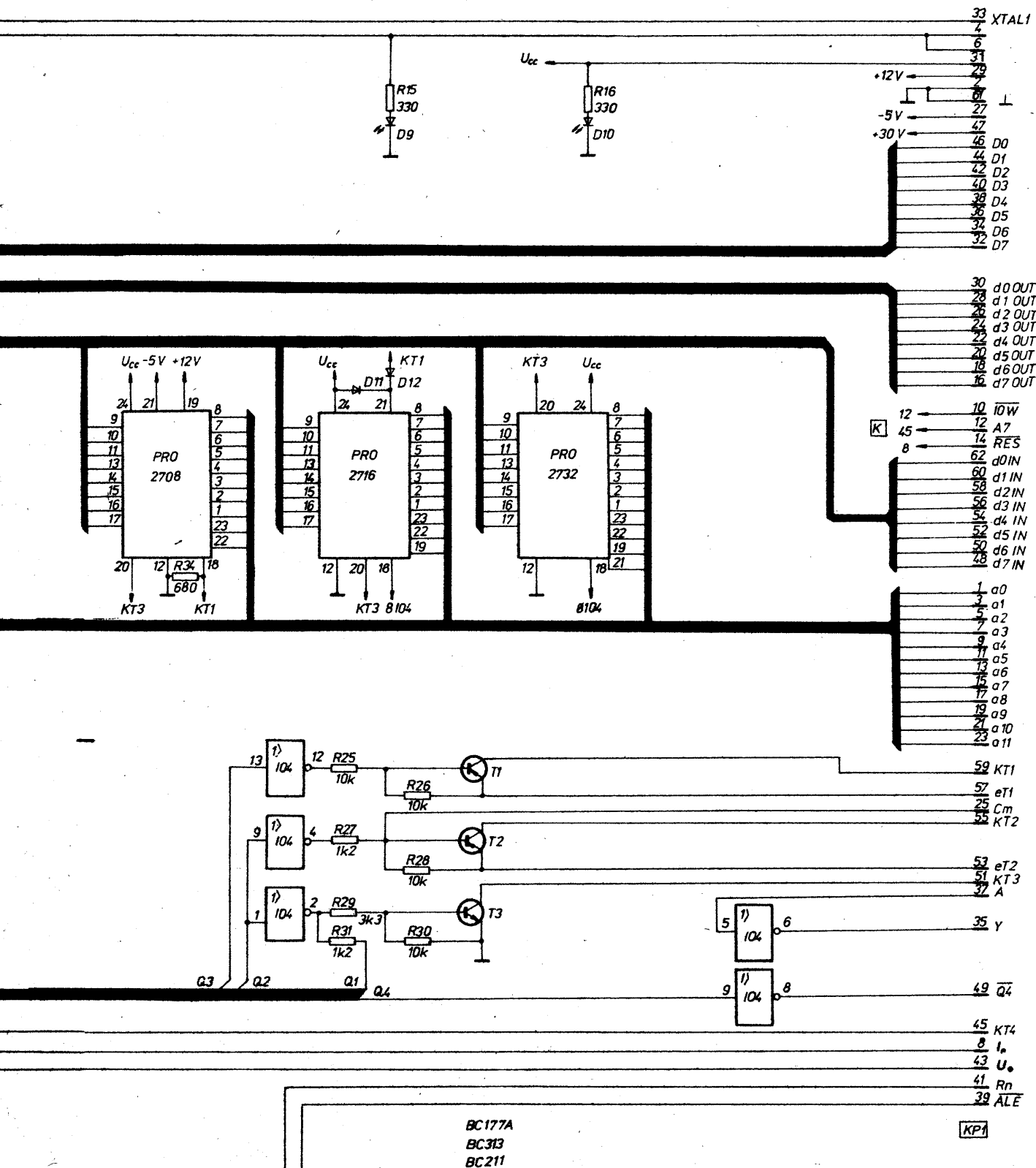
Seznam součástek

Rezistory

R1	100 Ω, TR 151
R2	220 Ω, TR 151
R3	100 Ω, TR 151
R4	470 Ω, TR 151
R5	10 kΩ, TR 191
R6	TR 191, 10 kΩ
R7	TR 191, 4,7 kΩ
R8	TR 191, 2,2 kΩ
R9	TR 191, 4,7 kΩ
R10	TR 191, 10 kΩ
R11	TR 191, 4,7 kΩ
R12	TR 151, 22 Ω
R13	TR 191, 18 kΩ
R14	TR 191, 4,7 kΩ

R15	TR 151, 330 Ω
R16	TR 151, 330 Ω
R17	TR 191, 4,7 kΩ
R18	TR 191, 4,7 kΩ
R19	TR 191, 4,7 kΩ

R20	TR 191, 4,7 kΩ
R21	TR 191, 4,7 kΩ
R22	TR 191, 4,7 kΩ
R23	TR 191, 4,7 kΩ
R24	TR 191, 4,7 kΩ



R25 TR 191, 22 Ω
R26 TR 191, 10 Ω
R27 TR 191, 1,2 Ω
R28 TR 191, 19 Ω
R29 TR 191, 3,3 Ω
R30 TR 191, 10 Ω
R31 TR 191, 1,2 Ω
R32 TR 191, 1 Ω
R33 TR 191, 1 Ω
R34 TR 152, 680 Ω

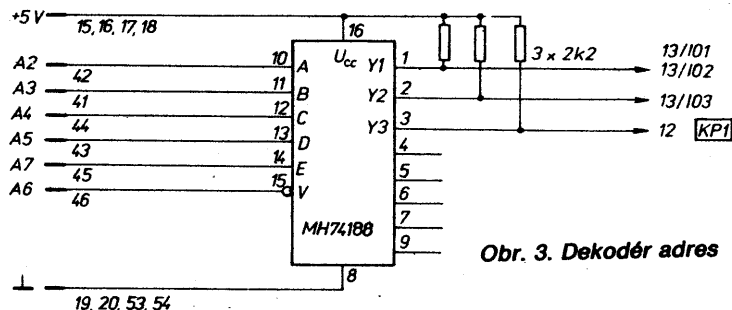
a jeho činnost. Není-li přítomen signál ALE, nesepe se programovací napětí ze stabilizátoru. Programování neúplně zasunutého obvodu by vedlo k jeho poškození.

Adresové signály vznikají na čítačích IO5, IO6 a IO7. Vstup kaskády čítačů a nulování se ovládají z IO3. Toto řešení je výhodné (pouze dva bity na ovládání) při adresování, které jde vzestupně. Pokud bychom chtěli někdy číst (programovat) jen určité adresy, musíme do programu zvyšujícího adresu vložit pomocné počítadlo, které nastavíme na požadovanou adresu a vstup dostane počet impulsů, potřebný k sestavení žádné adresy.

Vhodně časované signály požadovaných úrovní se řídí přes IO3 programem. Pouze ovládání T2 a T3 je zabezpečeno obvodově, protože jejich kolektory jsou u některých pamětí spojeny a při nesprávném programu by mohlo dojít k jejich současnému sepnutí. To by vedlo (např. u 2708) k výpadku +12 V a narušení činnosti celého mikropočítače.

Tab. 2b. Nastavení bitů IO3 a IO101 pro některé typy pamětí.

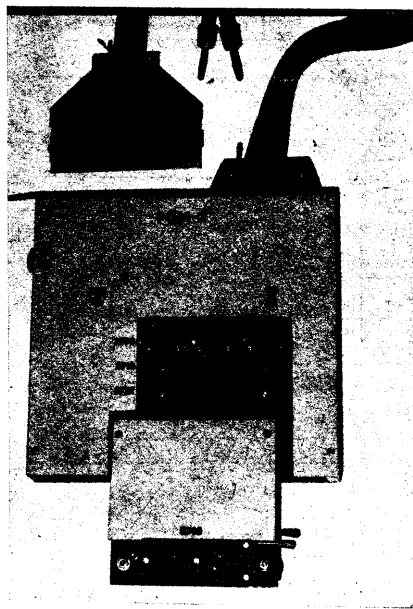
paměť bit	b7	b6	b5	b4	b3	b2	b1	b0	
2708									
blokování	L	X	L	L	X	PGM	WE	CE	IO3
čtení	L	L	H	1	X	L	L	L	
programování	H	H	H	1	X	1	H	L	
2716									
blokování	L	X	L	L	CE	Vpp	L	OE	IO3
čtení	L	L	H	1	L	L	L	L	
programování	H	H	H	1	1	H	L	L	
2732									
blokování	L	X	L	L	CE	X	Vpp	OE	IO3
čtení	L	L	H	1	L	X	L	X	
programování	H	H	H	1	1	X	H	L	
PROM									
blokování	Ucc1	X	L	L	CS	X	Ucc2	X	IO3
čtení	H	H	H	1	L	X	L	X	
programování	H	H	H	1	1	X	H	X	
8748									
zápis adresy	H	L	H	L	X	EA	PROG	PROG	IO3
zápis dat	L	L	X	X	X	T0	RES	VDD	IO101
programování	H	H	X	X	X	H	L	L	IO3
čtení	L	L	X	1	X	H	H	H	IO101
	L	L	H	1	X	L	L	L	IO3
	L	L	X	X	X	L	H	L	IO101



Obr. 3. Dekodér adres

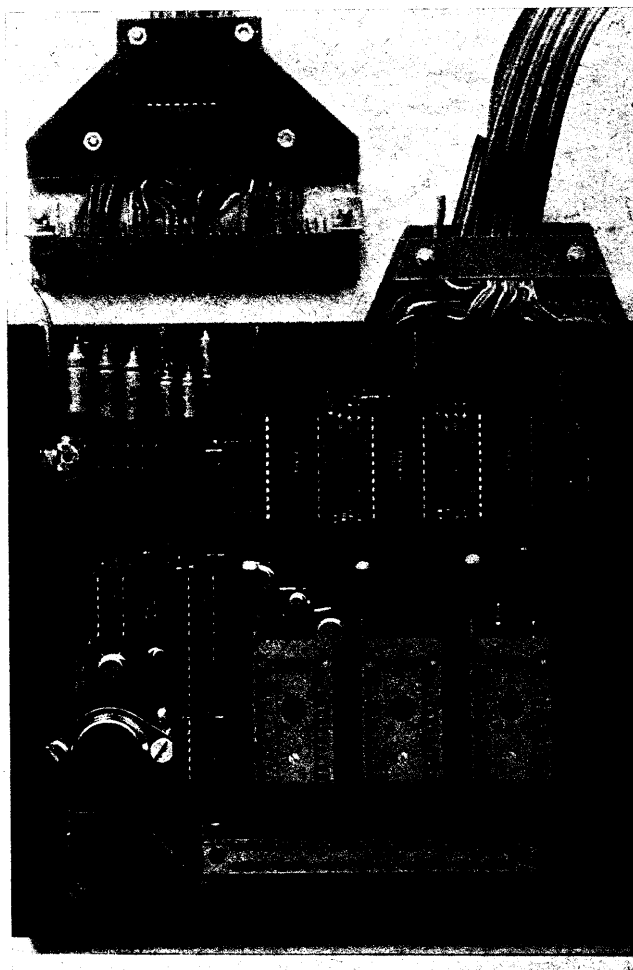
navrhnout plošné spoje na počítači se systémem CAD, kde se práce spojené s návrhem a tvorbou DPS výrazně zkrátí, je časově výhodnější použít univerzální desku.

Všechny paměti se měly původně připojovat jako moduly do konektoru KP-1. Na desce však nebyla využita celá plocha a tak jsem pro nejvíce používané paměti 2708, 2716 a 2732 umístil objímky na základní desku. Tím je možné zjednodušit programátor (pracuje-li se jen s těmito typy pamětí) tak, že se neosadí konektor KP-1

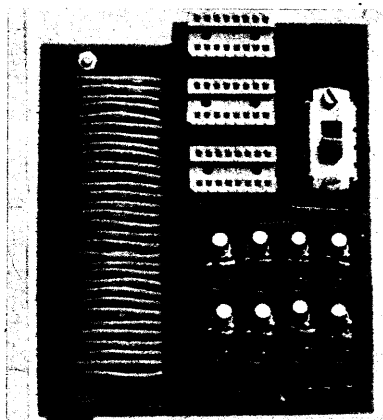
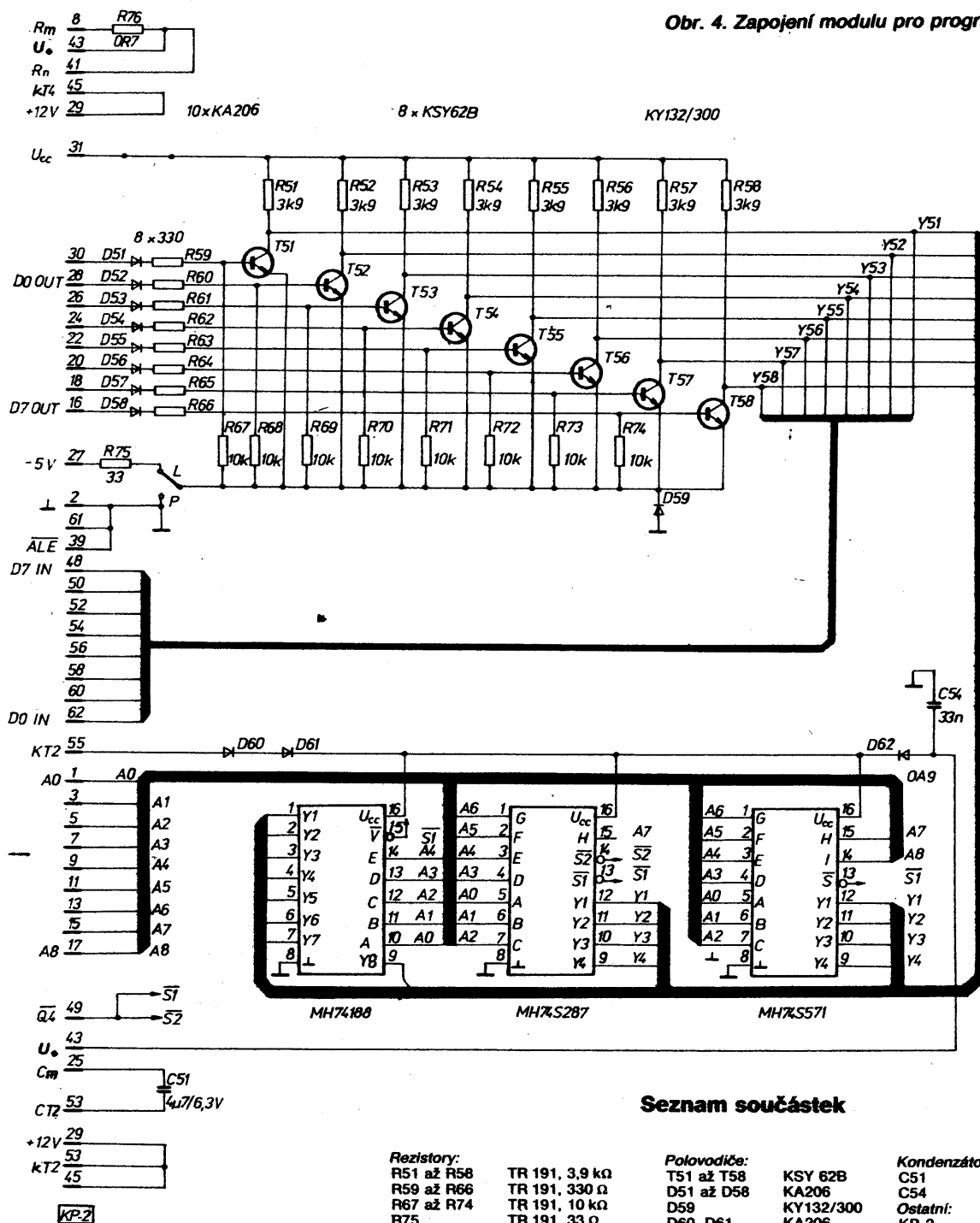


Celkový vzhled a vnitřní uspořádání programátoru PGM-2

PGM-2 byl sestaven na desce BDS-1. Navrhnout (dobře) plošný spoj pro hustotu součástek a spojů, jaký je v programátoru, je práce vyžadující mnoho času. Pokud nebude možné



Obr. 4. Zapojení modulu pro programování PROM



Vzhled modulu pro programování PROM

a propojky (nutné pro funkci programování) se provedou drátovými spojkami podle zapojení modulu EPROM.

K základní desce je zhotoven modul PROM pro paměti 74188, 74S287, 74S571, modul EPROM s nutnými propojkami pro programování a modul mikropočítače 8748.

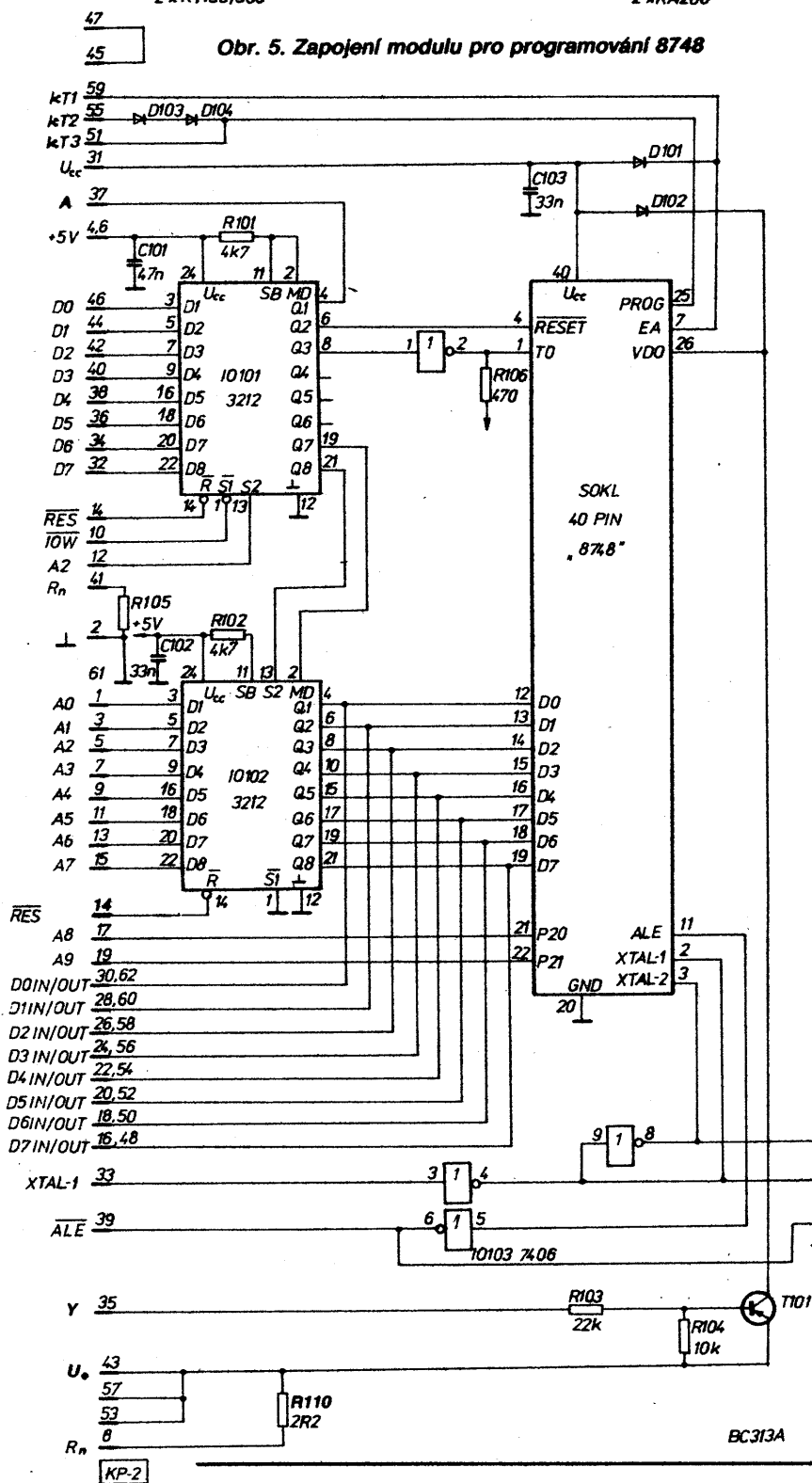
Programové vybavení

Údaje potřebné pro psaní programu jsou v tab. 2. Programy byly psány tak, aby nebyly delší než 1 kB a zajistily všechny nutné operace s pamětí.

Obvod 74188 je se svými 256 bity pamětí nejmenší, ale vyžaduje poměrně komplikované programování. (Viz vývojový diagram na str. 59).

Programátor PGM-2 byl nejprve řízen SAPI-1 s deskou REM-1 (SRAM+ Mikromonitor). Pro desku DRAM 48 kB a Monitor V.4.0 byl řídicí program upraven. Ovládáme jej podle tabulky služeb — tab. 3. Programy se spouštějí příkazem G u Monitoru V.4.0 (u Mikro S). Po provedení příkazu se vypíše údaj, který nás informuje o výsledku. Je to adresa první chyby právě prováděné operace. U kontroly čistoty je to adresa prvního bajtu, kde není požadovaný obsah (u PROM a 8748 00H, u EPROM FF H). Při programování se vypíše první adresa, kde nedošlo k naprogramování dat, u čtení první adresa, která se již

Obr. 5. Zapojení modulu pro programování 8748



Tab. 3a. Tabulka služeb pro Monitor V4.0.

	Čtení dat	Kontrola čistoty	Programování
74 188	1840 H	188E H	1800 H
74S287	184E H	18A0 H	1824 H
74S571	185C H	18AC H	1832 H
2708	186A H	18B8 H	18EB H
2716	1876 H	18D3 H	1846 H
2732	1882 H	18DF H	197F H
8748	19B8 H	18C7 H	19FF H

Tab. 3b. Tabulka služeb pro Mikromonitor.

	Čtení dat	Kontrola čistoty	Programování
74 188	1846 H	1897 H	1809 H
74S287	1854 H	18A6 H	182A H
74S571	1862 H	18B2 H	1838 H
2708	1870 H	18BE H	18F4 H
2716	187C H	18D9 H	194C H
2732	1888 H	18E5 H	1985 H
8748	19BE H	18CD H	1A05 H

nečetla. U programování paměti PROM se ještě vypíše počet bitů, které byly programovány na třetí pokus programovacím pulsem 20 ms (normální $t=1$ ms). Zobrazení je jen u Monitoru V4.0. U Mikro musíme údaj adresy vyvolat dodatečně. Je uložen na 9FFE a 9FFF H, údaj třetích pokusů PROM je na adrese 9FFD H.

Příklad: po kontrole čistoty bude u dobrého 74S287 výpis 0100 a u 2716 výpis 0800. Po úspěšném programování 74188 bude výpis 00 0020; dojde-li k chybě při programování, vypíše se např. 03 0011 (3 bity programovány na 3. pokus a neúspěšné programování, chyba je na adrese 0011 H). U správně naprogramované 2708 bude výpis 0400.

Jsou určeny pevné adresy, kam se začínají ukládat načtená data (A000 H) a kde jsou uložena data k programování (C000 H). Řídící program, přeložený do adresy 1800 H, je umístěn v 6. pozici EPROM desky REM-1.

Seznam součástek

Rezistory:

R101, R102	TR 181, 4,7 kΩ
R103	TR 191, 22 kΩ
R104	TR 191, 10 kΩ
R105	
R105 až R109	TR 191, 470 Ω

Kondenzátory:

C101	TK783, 47 nF
C102, C103	TK783, 33 nF

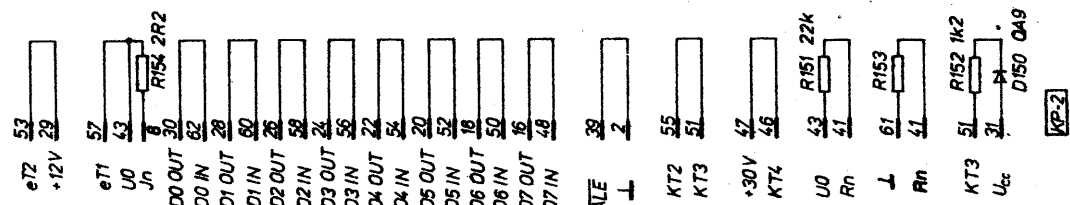
Polovodiče:

D101, D102	KA206
D103, D104	KY130/300
T101	BC313A
IO1, IO2	MH3212
IO3	UCY7406

Ostatní:
KP-2

konektor FRB TY 517 62 12

Obr. 6. Zapojení propojek pro programování paměti EPROM



Seznam součástek

Rezistory:

R151	TR191, 22 kΩ
R152	TR191, 1,2 kΩ
R153	TR191
R154	2,2 Ω

Polovodiče:

D150	OA9
------	-----

Ostatní:

KP-2

konektor FRB TY 517 62 12

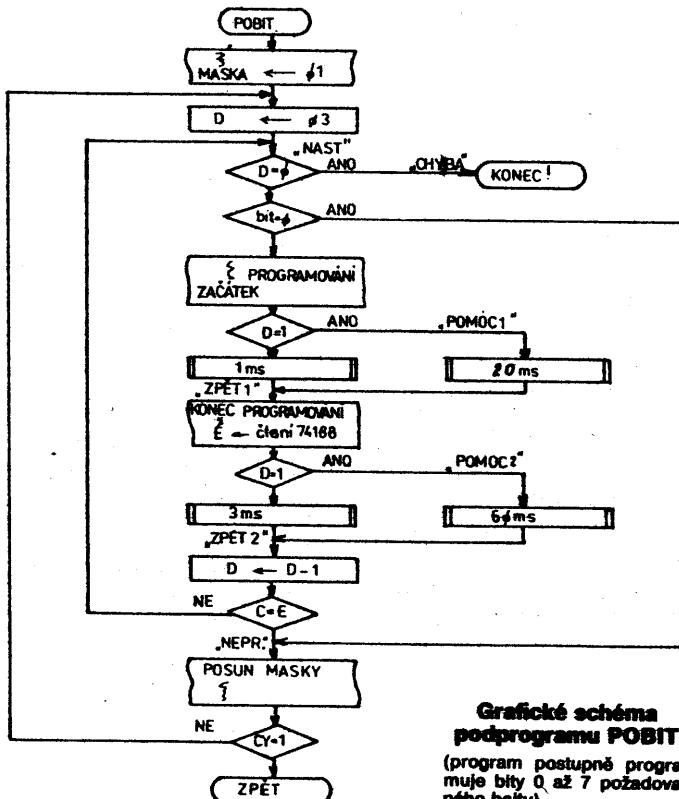
Řídicí program lze jednoduše modifikovat s ohledem na pevně určené adresy dat čtení/programování, adresy instrukcí IN/OUT i místa, kam se řídicí program uloží změnou direktiv v assemblerovském programu. Tím lze jednoduše přizpůsobit PGM-2 pro jiné požadavky u SAPI 1, nebo i jiný řídicí mikropočítač.

Závěr

Programovací zařízení PGM-2 řeší programování pamětí technicky jednoduše a modulární uspořádání umožňuje jednoduché rozšíření na nejrůznější typy pamětí.

Ke každému programátoru pamětí EPROM patří také zařízení pro mazání těchto pamětí. Používá se ultrafialové světlo s vlnovou délkou 2537 Å (Angström). Zdroj světla musí být ve vzdálenosti 25 mm od okénka pamětí. Doba pro vymazání je dána typem výbojky; u pamětí fy Intel je k vymazání zapotřebí energie 15 Ws/cm². Příspěvek vznikl na podzim r. 1985. Dnes je programátor PGM-2 na deskách s plošnými spoji a pracuje i pod operačním systémem CP/M s komfortnějším programovým vybavením.

Grafické schéma obslužného programu



Grafické schéma podprogramu POBIT (program postupně programuje bity 0 až 7 požadovaného bajtu)

POBIT - NULOVACÍ PROGRAM PRO PAMĚTI PGM-2

*1800	CD 09 18 CD AD 1B C3 26	*1968	EC D3 08 3E A9 D3 08 CD	*1AD0	CD 22 1B AF D3 04 CD 12
*1808	05 11 20 00 3E 08 32 FC	*1970	EA 1B 3E B0 D3 08 CD 67	*1AD8	1B 3E E8 D3 08 CD 22 1B
*1810	9F CD 7C 1A 7E CD 08 1A	*1978	1A C2 56 19 C3 6C 1A CD	*1AE0	DB 04 5F 3E 68 D3 08 7A
*1818	CD 73 1A CD 67 1A C2 14	*1980	85 19 C3 EE 18 21 00 C0	*1AE8	FE 01 CA 53 18 CD 30 1B
*1820	18 C3 6C 1A CD 2A 18 C3	*1988	11 00 10 3E C2 D3 08 3E	*1AF0	15 7B A1 B9 C2 92 1A 79
*1828	03 18 11 00 01 3E 1C C3	*1990	E2 D3 08 7E D3 04 CD 12	*1AF8	07 4F F5 3A FC 9F FE 88
*1830	0E 18 CD 38 18 C3 03 18	*1998	1B 3E EA D3 08 CD 3E 1B	*1B00	CA 08 1B F1 FE 10 C2 90
*1838	11 00 02 3E 57 C3 0E 18	*19A0	3E E2 D3 08 3E A9 D3 08	*1B08	1A D1 C9 F1 D2 90 1A C3
*1840	CD 46 18 C3 26 05 06 88	*19A8	CD EA 1B 3E B2 D3 08 CD	*1B10	09 1B D5 11 01 00 CD 4C
*1848	11 20 00 C3 6C 1B CD 54	*19B0	67 1A C2 8F 19 C3 6C 1A	*1B18	1B D1 C9 D5 11 05 00 C3
*1850	18 C3 26 05 06 28 11 00	*19B8	CD BE 19 C3 EE 18 CD F8	*1B20	16 1B D5 11 0A 00 C3 16
*1858	01 C3 6C 1B CD 62 18 C3	*19C0	1B 3E 80 06 04 CD F6 19	*1B28	1B D5 11 50 00 C3 16 1B
*1860	26 05 06 57 11 00 02 C3	*19C8	3E A4 06 04 CD F6 19 06	*1B30	D5 11 00 02 C3 16 1B D5
*1868	6C 1B CD 70 18 C3 26 05	*19D0	C4 CD F8 19 06 C6 CD F8	*1B38	11 70 06 C3 16 1B D5 11
*1870	11 00 04 C3 BD 1B CD 7C	*19D8	19 06 06 CD F8 19 3E B4	*1B40	00 11 C3 16 1B D5 11 00
*1878	18 C3 26 05 11 00 08 C3	*19E0	06 02 CD F6 19 D8 04 77	*1B48	13 C3 16 1B 1B 7A B3 C2
*1880	73 18 CD 88 18 C3 26 05	*19E8	06 04 CD F8 19 CD 67 1A	*1B50	4C 1B C9 CD 45 1B 3A FD
*1888	11 00 10 C3 73 18 CD 97	*19F0	C2 C8 19 C3 6C 1A D3 08	*1B58	9F 3C 32 FD 9F C3 F0 1A
*1890	18 CD 9C 1B C3 26 05 11	*19F8	78 D3 0C CD 12 1B C9 CD	*1B60	CD 37 1B C3 C5 1A D1 33
*1898	20 00 06 00 CD 09 1B C9	*1A00	05 1A C3 03 18 CD F1 18	*1B68	33 C3 6C 1A 21 00 A0 3E
*18A0	CD A6 18 C3 91 18 11 00	*1A08	3E 80 06 00 CD F6 19 3E	*1B70	80 D3 08 3E 88 D3 08 CD
*18A8	01 C3 9A 18 CD B2 18 C3	*1A10	A0 06 04 CD F6 19 3E A4	*1B78	37 1B D8 04 4F 78 FE 88
*18B0	91 18 11 00 02 C3 9A 18	*1A18	CD F6 19 06 C4 CD F6 19	*1B80	CA 98 1B 79 E6 0F 77 3E
*18B8	CD 8E 18 C3 91 18 11 00	*1A20	06 C6 CD F6 19 06 06 CD	*1B88	B8 D3 08 3E A8 D3 08 CD
*18C0	04 06 FF CD 09 1B C9 CD	*1A28	F6 19 3E E5 CD F6 19 7E	*1B90	67 1A C2 7A 1B C3 6C 1A
*18C8	CD 18 C3 91 18 11 00 04	*1A30	D3 04 3E E5 06 07 CD F6	*1B98	79 C3 86 1B 2A FE 9F 7C
*18D0	C3 9A 18 CD 09 18 C3 91	*1A38	19 3E E6 CD F6 19 CD 45	*1BA0	E6 0F 67 CD 46 06 CD 24
*18D8	18 11 00 08 C3 C1 18 CD	*1A40	1B 3E E5 CD F6 19 06 06	*1BA8	01 CD 24 01 C9 3A FD 9F
*18E0	E5 18 C3 91 18 11 00 10	*1A48	CD F6 19 3E A4 06 02 CD	*1BB0	CD 33 01 CD 3C 01 CD 3C
*18E8	C3 C1 18 CD F4 18 CD 9C	*1A50	F6 19 CD EA 1B 3E B4 06	*1BB8	01 CD 9C 1B C9 21 00 A0
*18F0	1B C3 26 05 01 40 01 CD	*1A58	04 CD F6 19 3E A4 CD 67	*1BC0	3E 09 D3 08 D8 04 77 3E
*18F8	F1 1B 3E 82 D3 08 3E E2	*1A60	1A C2 13 1A C3 6C 1A 23	*1BC8	39 D3 08 3E 29 D3 08 CD
*1900	D3 08 7E D3 04 CD 12 1B	*1A68	1B 7A B3 C9 22 FE 9F AF	*1BD0	67 1A C2 C4 1B AF D3 08
*1908	3E E6 D3 08 CD 22 1B 3E	*1A70	D3 08 C9 3E 78 D3 08 3E	*1BD8	C9 21 00 A0 7E B8 C2 6C
*1910	E2 D3 08 3E F2 D3 08 CD	*1A78	68 D3 08 C9 21 00 C0 AF	*1BE0	1A CD 67 1A C2 DC 1B C3
*1918	67 1A C2 FE 18 08 78 B1	*1A80	32 FD 9F 3E 48 D3 08 C9	*1BE8	6C 1A D8 04 BE C2 67 1B
*1920	C2 F7 18 AF D3 08 11 00	*1A88	D5 47 3E 68 D3 08 0E 01	*1BF0	C9 21 00 C0 11 00 04 C9
*1928	00 CD 4C 1B 21 00 C0 11	*1A90	16 03 7A FE 00 CA 66 1B	*1BF8	21 00 A0 C3 F4 1B FF FF
*1930	00 04 3E 21 D3 08 CD EA	*1A98	78 A1 CA F7 1A 3E E8 D3		
*1938	1B 3E 31 D3 08 CD 67 1A	*1AA0	08 CD 12 1B 3E E0 D3 08		
*1940	C2 32 19 C3 6C 1A CD 4C	*1AA8	CD 12 1B 79 D3 04 CD 12		
*1948	19 C3 EE 18 21 00 C0 11	*1AB0	1B 3E E2 D3 08 CD 22 1B		
*1950	00 08 3E CC D3 08 3E EC	*1AB8	3E EA D3 08 7A FE 01 CA		
*1958	D3 08 7E D3 04 CD 12 1B	*1AC0	60 1B CD 29 1B 3E E2 D3		
*1960	3E E4 D3 08 CD 3E 1B 3E	*1AC8	08 CD 12 1B 3E E0 D3 08		

Kapky matematiky v elektrotechnice

Ing. Lumír Dvořák

Souvislosti a velikosti fyzikálních hodnot součástek elektrických obvodů, průběhy a závislosti napětí a proudů v nich, to vše je dohromady vázáno základními matematickými vztahy. Měli by je znát i ti „nejzarputilejší“ praktikové, protože umožňují pochopit, co se v elektrických obvodech děje, proč se to děje, a jak to můžeme změnou jednotlivých součástí obvodu ovlivnit.

Proto jsme zařadili do této ročenky i tak nezvyklý a neobvykle zpracovaný příspěvek, „Kapky matematiky v elektrotechnice“. Necht' vám všem poslouží k neustálému opakování a obnovování znalostí těch základních zákonitostí elektrických obvodů, s kterými přicházíte denně do styku.

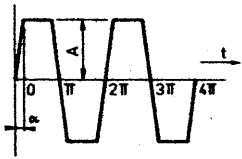
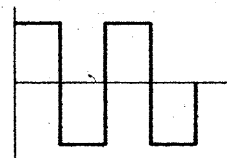
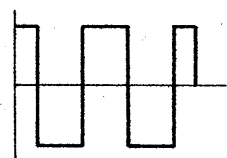
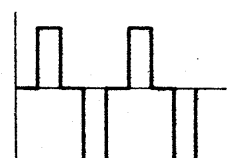
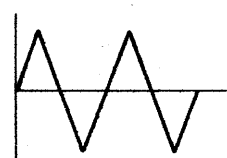
Základním pojmem při popisu periodických průběhů napětí a proudů v elektrických obvodech je goniometrická Fourierova řada. Platí, že periodickou funkci $f(t)=f(t+2k\pi)$, kde k je přirozené číslo s periodou 2π , která je v intervalu od 0 do 2π jednoznačná a konečná a má v tomto intervalu konečný počet nespojitostí, lze rozvinout v konečnou konvergentní goniometrickou Fourierovu řadu:

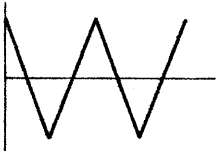
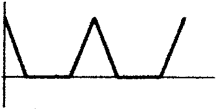
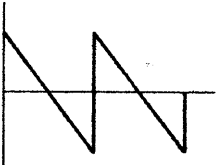
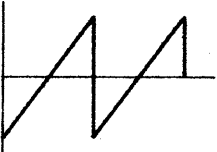
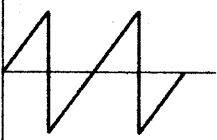
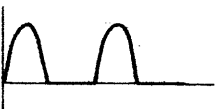
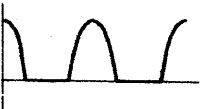
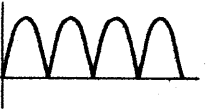

$$f(t) = a_0 + a_1 \cos t + b_1 \sin t + a_2 \cos 2t + b_2 \sin 2t + \dots + a_n \cos nt + b_n \sin nt$$

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(t) dt$$

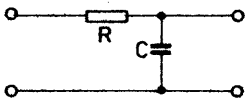
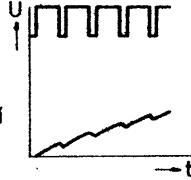
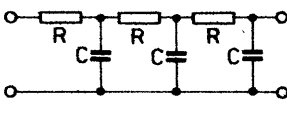
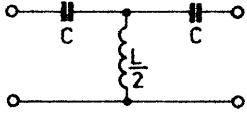
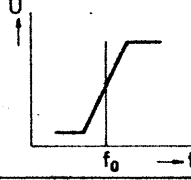
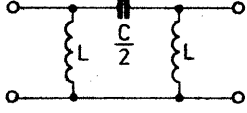
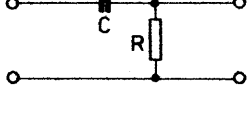
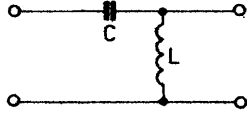
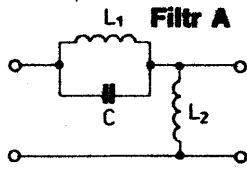
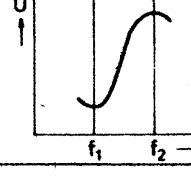
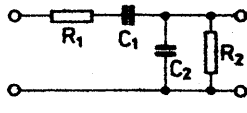
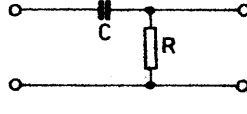
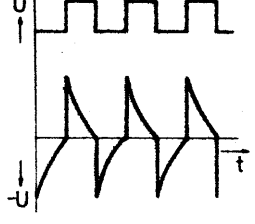
$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \cos kt dt$$

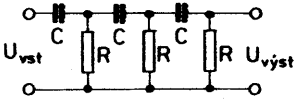
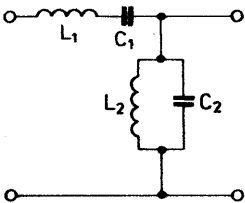
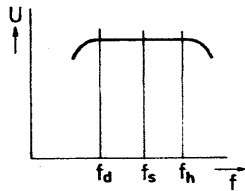
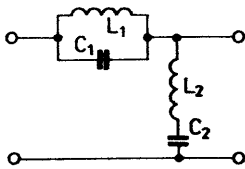
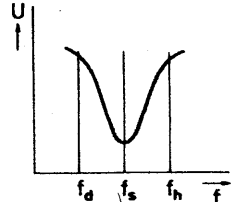
$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \sin kt dt$$

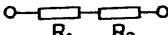

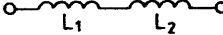
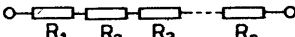
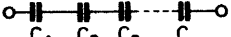

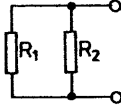
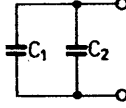
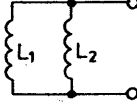
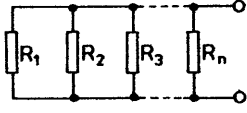
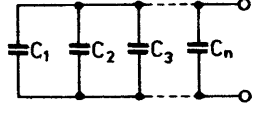
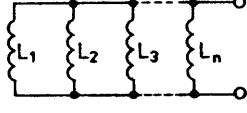
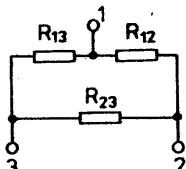
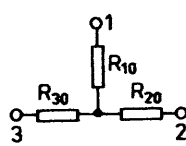
	<p>Lichoběžníkový průběh:</p> $f(t) = \frac{4A}{2\pi} \left(\frac{\sin \alpha}{1^2} \sin \omega t + \frac{\sin 2\alpha}{3^2} \sin 3\omega t + \frac{\sin 5\alpha}{5^2} \sin 5\omega t + \dots \right)$
	<p>Obdélníkový průběh:</p> $f(t) = \frac{4A}{\pi} \left(\sin \omega t + \frac{1}{3} \sin 3\omega t + \frac{1}{5} \sin 5\omega t + \dots \right)$
	<p>Obdélníkový průběh:</p> $f(t) = \frac{4A}{\pi} \left(\frac{\cos \alpha}{1} \cos \omega t + \frac{\cos 3\alpha}{3} \cos 3\omega t + \frac{\cos 5\alpha}{5} \cos 5\omega t + \dots \right)$
	<p>Obdélníkový průběh:</p> $f(t) = \frac{4A}{\pi} \left(\frac{\cos \alpha}{1} \sin \omega t + \frac{\cos 3\alpha}{3} \sin 3\omega t + \frac{\cos 5\alpha}{5} \sin 5\omega t + \dots \right)$
	<p>Trojúhelníkový průběh:</p> $f(t) = \frac{8A}{\pi^2} \left(\frac{1}{1^2} \sin \omega t - \frac{1}{3^2} \sin 3\omega t + \frac{1}{5^2} \sin 5\omega t - \dots \right)$

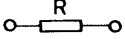
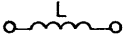
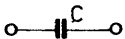
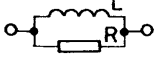
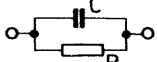


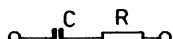


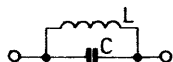
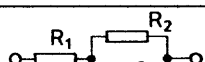
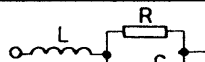
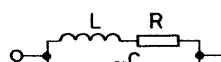
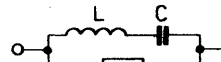
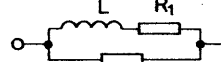
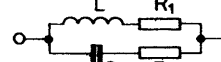
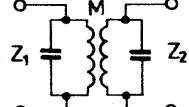
	$f(t) = \frac{8A}{\pi^2} \left(\frac{1}{1^2} \cos \omega t + \frac{1}{3^2} \cos 3\omega t + \frac{1}{5^2} \cos 5\omega t + \dots \right)$
	$f(t) = \frac{A\alpha}{2\pi} + \frac{2A}{\pi\alpha} \left(\frac{1-\cos\alpha}{1^2} \cos \omega t + \frac{1-\cos 2\alpha}{2^2} \cos 2\omega t + \frac{1-\cos 3\alpha}{3^2} \cos 3\omega t + \dots \right)$
	<p>Pilovitý průběh:</p> $f(t) = \frac{2A}{\pi} \left(\sin \omega t + \frac{1}{2} \sin 2\omega t + \frac{1}{3} \sin 3\omega t + \dots \right)$
	$f(t) = \frac{2A}{\pi} \left(\sin \omega t + \frac{1}{2} \sin 2\omega t + \frac{1}{3} \sin 3\omega t + \dots \right)$
	$f(t) = \frac{2A}{\pi} \left(\sin \omega t - \frac{1}{2} \sin 2\omega t + \frac{1}{3} \sin 3\omega t - \dots \right)$
	<p>Jednocestně usměrněný střídavý signál:</p> $f(t) = \frac{A}{\pi} \left(1 + \frac{\pi}{2} \cos \omega t - \frac{2}{3} \cos 2\omega t + \frac{2}{15} \cos 4\omega t - \dots \right)$
	$f(t) = \frac{A}{\pi} \left(1 + \frac{\pi}{2} \cos \omega t + \frac{2}{3} \cos 2\omega t - \frac{2}{15} \cos 4\omega t + \dots \right)$
	<p>Dvojcestně usměrněný střídavý signál:</p> $f(t) = \frac{2A}{\pi} \left(1 - \frac{2}{3} \cos 2\omega t - \frac{2}{15} \cos 4\omega t - \frac{2}{35} \cos 6\omega t - \dots \right)$
	$f(t) = \frac{3A}{\pi} \left(1 - \frac{2}{35} \cos 6\omega t - \frac{2}{143} \cos 12\omega t - \frac{2}{323} \cos 18\omega t - \dots \right)$

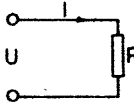
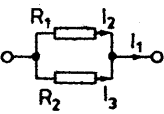
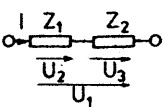
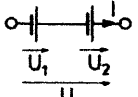
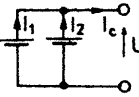
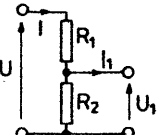
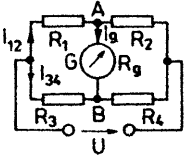
Dolní propusti	Článek T 	$L = \frac{Z}{2\pi f_0}$ $C = \frac{1}{2\pi f_0 Z}$ $Z = \sqrt{1-m^2}$ $m = \frac{f_2}{f_1}$ $f_0 = \frac{1}{2\pi \sqrt{LC}}$	
	Článek π 	$L = \frac{Z}{2\pi f_0}$ $C = \frac{1}{2\pi f_0 Z}$ $Z = \frac{1}{\sqrt{1-m^2}}$ $m = \frac{f_2}{f_1}$ $f_0 = \frac{1}{2\pi \sqrt{LC}}$	
	Dolní filtrační článek RC 	$R = \frac{1}{2\pi f_0 C}$ $C = \frac{1}{2\pi f_0 R}$ $f_0 = \frac{1}{2\pi RC}$	
	Dolní filtrační článek LC 	$L = \frac{Z}{2\pi f_0}$ $C = \frac{1}{2\pi f_0 Z}$ $Z = \sqrt{1-m^2}$ $m = \frac{f_2}{f_1}$ $f_0 = \frac{1}{2\pi \sqrt{LC}}$	
	Dolní filtr 	$L = \frac{mZ}{2\pi f_0}$ $C_1 = \frac{1-m^2}{m}$ $C_2 = \frac{m}{2\pi f_0 Z}$ $m = \sqrt{1-\left(\frac{f_2}{f_1}\right)^2}$	
Usměrňovací filtry LC	Jednoduchý filtr LC 	$L = \frac{10}{C} \frac{P_0}{p}$ $C = \frac{10}{L} \frac{P_0}{p}$ $LC = 10 \frac{P_0}{p}$	$L = \frac{2,5}{C} \frac{P_0}{p}$ $C = \frac{2,5}{L} \frac{P_0}{p}$ $LC = 2,5 \frac{P_0}{p}$
		$L = \frac{10}{C} \sqrt{\frac{P_0}{p}}$ $C = \frac{10}{L} \sqrt{\frac{P_0}{p}}$ $LC = \sqrt{\frac{P_0}{p}}$	$L = \frac{2,5}{C} \sqrt{\frac{P_0}{p}}$ $C = \frac{2,5}{L} \sqrt{\frac{P_0}{p}}$ $LC = 2,5 \sqrt{\frac{P_0}{p}}$
Usměrňovací filtry RC		$R = \frac{3000}{C} \frac{P_0}{p}$ $C = \frac{3000}{R} \frac{P_0}{p}$ $RC = 3000 \frac{P_0}{p}$ $U = U_0 - \frac{I_0 R}{1000} \quad [V; mA; \Omega]$	$R = \frac{1500}{C} \frac{P_0}{p}$ $C = \frac{1500}{R} \frac{P_0}{p}$
		$R = \frac{3000}{C} \sqrt{\frac{P_0}{p}}$ $C = \frac{3000}{R} \sqrt{\frac{P_0}{p}}$ $RC = 3000 \sqrt{\frac{P_0}{p}}$ $U = U_0 - \frac{2 I_0 R}{1000}$	$R = \frac{1500}{C} \sqrt{\frac{P_0}{p}}$ $C = \frac{1500}{R} \sqrt{\frac{P_0}{p}}$ $RC = 1500 \sqrt{\frac{P_0}{p}}$

Filtreační články RC	Integrační obvod 	$\tau = RC$ $u = U \left(1 - e^{-\frac{t}{\tau}}\right)$ u — okamžitá hodnota napětí při nabíjení U — velikost napájecího napětí τ — časová konstanta obvodu RC	
	Integrační obvod 	$R = \frac{\sqrt{6}}{2\pi f_0 C}$ $C = \frac{\sqrt{6}}{2\pi f_0 R}$ $f_0 = \frac{\sqrt{6}}{2\pi RC}$	
Horní propusti	Článek T 	$L = \frac{Z}{\pi f_0}$ $C = \frac{1}{2\pi f_0 Z}$ $f_0 = \frac{1}{2\pi \sqrt{LC}}$ $Z_{vst} = Z_{výst} = \sqrt{\frac{L}{C}}$	
	Článek TT 	$L = \frac{Z}{2\pi f_0}$ $C = \frac{1}{\pi f_0 Z}$ $f_0 = \frac{1}{2\pi \sqrt{LC}}$ $Z = \sqrt{\frac{L}{C}}$	
	Článek RC 	$R = \frac{1}{2\pi f_0 C}$ $C = \frac{1}{2\pi f_0 R}$ $f_0 = \frac{1}{2\pi RC}$	
	Článek LC 	$L = \frac{Z}{2\pi f_0}$ $C = \frac{1}{2\pi f_0 Z}$ $f_0 = \frac{1}{2\pi \sqrt{LC}}$ $Z = \sqrt{\frac{L}{C}}$	
	Filtr A 	$L_1 = \frac{m}{1-m^2} \frac{Z}{2\pi f_0}$ $L_2 = \frac{1}{m} \frac{1}{2\pi f_0}$ $C_1 = \frac{1}{m} \frac{1}{2\pi f_0 Z}$ $m = \sqrt{1 - \left(\frac{f_2}{f_1}\right)^2}$ $Z = \sqrt{\frac{L}{C}}$	
	Filtr B 	$R_1 = R_2 = \frac{1}{2\pi f_0 C} = R$ $C_1 = C_2 = \frac{1}{2\pi f_0 R} = C$ $f_0 = \frac{1}{2\pi RC}$	
	Derivační obvod 	$\tau = RC$ $u = U e^{-\frac{t}{\tau}}$ u — okamžitá hodnota napětí U — amplituda impulsů τ — časová konstanta obvodu RC	

	Trojčlankový filtr RC 	$f_0 = \frac{1}{2\pi RC\sqrt{6}}$ $\frac{U_{vyst}}{U_{vst}} = -\frac{1}{29}$
	Pásmový filtr LC 	$f_s = \frac{1}{2\pi\sqrt{L_1 C_1}} = \sqrt{f_d f_h}$ $f_d = \sqrt{\frac{1}{4C_2 L_1} + \frac{1}{C_1 L_1}} - \frac{1}{2\sqrt{C_2 L_1}} \cdot \frac{1}{2\pi}$ $f_h = \sqrt{\frac{1}{4C_2 L_1} + \frac{1}{C_1 L_1}} + \frac{1}{2\pi\sqrt{C_2 L_1}} \cdot \frac{1}{2\pi}$ $L_1 = \frac{Z}{2\pi\dot{s}} \quad C_1 = \frac{\dot{s}}{2\pi Z f_d f_h} \quad \dot{s} = f_h - f_d$ $L_2 = \frac{Z\dot{s}}{2\pi(f_d + f_h)} \quad C_2 = \frac{1}{2\pi Z\dot{s}} \quad Z_{vst} = Z_{vyst} = Z$ 
	Pásmová zadrž LC 	$f_s = \frac{1}{2\pi\sqrt{L_1 C_1}} = \frac{1}{2\pi\sqrt{L_2 C_2}} = \sqrt{f_h f_d}$ $L_1 = \frac{Z\dot{s}}{2\pi f_d f_h} \quad C_1 = \frac{1}{2\pi Z\dot{s}}$ $L_2 = \frac{Z}{2\pi\dot{s}} \quad C_2 = \frac{1}{2\pi Z f_d f_h}$  $Z_{vst} = Z_{vyst} = Z$

Sériové řazení	rezistory  $R_S = R_1 + R_2$	kondenzátory  $C_S = \frac{C_1 C_2}{C_1 + C_2}$	cívky  $L_S = L_1 + L_2$	
	 $R_S = R_1 + R_2 + R_3 + \dots + R_n$	 $\frac{1}{C_S} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \dots + \frac{1}{C_n}$	 $L_S = L_1 + L_2 + L_3 + \dots + L_n$	
Paralelní řazení	rezistory  $R_P = \frac{R_1 R_2}{R_1 + R_2}$	kondenzátory  $C_P = C_1 + C_2$	cívky  $L_P = \frac{L_1 L_2}{L_1 + L_2}$	
	 $\frac{1}{R_P} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots + \frac{1}{R_n}$	 $C_P = C_1 + C_2 + C_3 + \dots + C_n$	 $\frac{1}{L_P} = \frac{1}{L_1} + \frac{1}{L_2} + \frac{1}{L_3} + \dots + \frac{1}{L_n}$	
hvězda — trojúhelník	z trojúhelníku do hvězdy  $R_{10} = \frac{R_{12} R_{13}}{R_{12} + R_{13} + R_{23}}$ $R_{20} = \frac{R_{12} R_{23}}{R_{12} + R_{13} + R_{23}}$ $R_{30} = \frac{R_{13} R_{23}}{R_{12} + R_{13} + R_{23}}$		z hvězdy do trojúhelníku  $R_{12} = R_{10} R_{20} \left(\frac{1}{R_{10}} + \frac{1}{R_{20}} + \frac{1}{R_{30}} \right)$ $R_{13} = R_{10} R_{30} \left(\frac{1}{R_{10}} + \frac{1}{R_{20}} + \frac{1}{R_{30}} \right)$ $R_{23} = R_{20} R_{30} \left(\frac{1}{R_{10}} + \frac{1}{R_{20}} + \frac{1}{R_{30}} \right)$	

	úhel mezi U a I	Impedance obvodu Z
	$\varphi = 0$	$Z = R$
	$\varphi = +90^\circ$	$Z = \omega L$
	$\varphi = -90^\circ$	$Z = \frac{1}{\omega C}$
	$\operatorname{tg} \varphi = \frac{\omega L}{R}$	$Z = \sqrt{\frac{R^2 \omega^2 L^2}{R^2 + \omega^2 L^2}}$
	$\operatorname{tg} \varphi = -\omega C R$	$Z = \sqrt{\frac{R^2}{1 + \omega^2 C^2 R^2}}$
	$\varphi = \pm 90^\circ$	$Z = \frac{\omega L}{1 - \omega^2 LC}$
	$\operatorname{tg} \varphi = \frac{\omega L}{R}$	$Z = \sqrt{R^2 + \omega^2 L^2}$
	$\operatorname{tg} \varphi = -\frac{1}{\omega C R}$	$Z = \sqrt{R^2 + \frac{1}{\omega^2 C^2}}$
	$\varphi = \pm 90^\circ$	$Z = \omega L - \frac{1}{\omega C}$
	$\operatorname{tg} \varphi = \frac{\omega L - \frac{1}{\omega C}}{R}$	$Z = \sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2}$
	$\operatorname{tg} \varphi = R \left(\frac{1}{\omega L} - \omega C \right)$	$Z = \sqrt{\frac{1}{\frac{1}{R^2} + \left(\omega C - \frac{1}{\omega L}\right)^2}}$
	$-\operatorname{tg} \varphi = \frac{\omega C R_2^2 (1 + \omega^2 C^2 R_2^2)}{R_1 (1 + \omega^2 C^2 R_2^2) + R_2}$	$Z = \sqrt{\left(R_1 + \frac{R_2}{1 + (\omega C R_2)^2}\right)^2 + \left(\frac{\omega C R_2^2}{1 + (\omega C R_2)^2}\right)^2}$
	$\operatorname{tg} \varphi = R \left[\omega L \left(\frac{1}{R^2} + \omega^2 L^2 \right) - \omega C \right]$	$Z = \sqrt{\omega^2 L^2 + \frac{1 - 2\omega^2 LC}{R^2 + \omega^2 C^2}}$
	$\operatorname{tg} \varphi = \frac{\omega [L(1 - \omega^2 LC) - CR^2]}{R}$	$Z = \sqrt{\frac{R^2 + \omega^2 L^2}{(1 - \omega^2 LC)^2 + \omega^2 C^2 R^2}}$
	$\operatorname{tg} \varphi = \frac{R}{\omega L - \frac{1}{\omega C}}$	$Z = \sqrt{\frac{R^2 \left(\omega L - \frac{1}{\omega C}\right)^2}{R^2 \left(\omega L - \frac{1}{\omega C}\right)^2}}$
	$\operatorname{tg} \varphi = \frac{L R_2}{R_1 (R_1 + R_2) + \omega^2 L^2}$	$Z = \sqrt{\frac{R_1^2 + \omega^2 L^2}{(R_1 + R_2)^2 + \omega^2 L^2}}$
	$\varphi = 0$	$Z = R_1 = R_2$
	$\varphi = \pm 180^\circ$	$Z = \sqrt{Z_1 Z_2} \quad Z_1 = \frac{\omega^2 M^2}{Z_2}$

Ohmův zákon 	stejnoseměrný obvod $U = RI$ [V; Ω; A] $I = \frac{U}{R}$ $R = \frac{U}{I}$	střídavý obvod $U = ZI$ $I = \frac{U}{Z}$ $Z = \frac{U}{I}$	
Výkon	stejnoseměrného proudu $P = UI$ [W; V; A] $P = RI^2 = \frac{U^2}{R}$ $U = \sqrt{PR}$ $I = \sqrt{\frac{P}{R}}$	střídavého proudu $P_s = UI$ [VA; V; A] $P = UI \cos \varphi$ [W; V; A] $P_q = UI \sin \varphi$ [VAR; V; A] φ	
Střídavý proud a střídavé napětí	efektivní hodnota $U = \frac{U_m}{\sqrt{2}} = 1,11 U_s$ $I = \frac{I_m}{\sqrt{2}} = 1,11 I_s$	střední hodnota $U_s = \frac{2U_m}{\pi} = \frac{2\sqrt{2}}{\pi} U$ $I_s = \frac{2I_m}{\pi} = \frac{2\sqrt{2}}{\pi} I$	okamžitá hodnota $u = U_m \sin \omega t$ $i = I_m \sin(\omega t + \varphi)$ $\omega = 2\pi f$
Kirchhoffův zákon	první:  $\sum I = 0$ $I_1 = I_2 + I_3$ $I_2 + I_3 - I_1 = 0$	druhý:  $\sum U = \sum RI = 0$ $U_1 = U_2 + U_3$ $Z_1 I + Z_2 I - U_1 = 0$	
Reaktance	kondenzátor $X_c = \frac{1}{\omega C} = \frac{1}{2\pi f C}$ [Ω; s ⁻¹ ; F]	cívka $X_L = \omega L = 2\pi f L$ [Ω; s ⁻¹ ; H]	rezistor $X_R = R$ [Ω]
Rezonanční kmitočet	$f_0 = \frac{1}{2\pi \sqrt{LC}}$ [Hz; H; F] $f_0^2 = \frac{25\,330}{LC}$ [MHz; μH; pF]		
Zdroje	sériové spojení  $U_e = U_1 + U_2 + \dots + U_n$	paralelní spojení  $I_c = I_1 + I_2 + \dots + I_n$	
Dělič napětí 	nezatížený dělič $I_1 = 0$ $U_1 = U \frac{R_2}{R_1 + R_2} = RI_2$	zatížený dělič $I_1 > 0$ $U_1 = U \frac{R_2}{R_1 + R_2} - I \frac{R_1 R_2}{R_1 + R_2}$	
Můstek 	vyvážený $U_{AB} = 0$ $\frac{R_1}{R_2} = \frac{R_3}{R_4}$ $R_1 = \frac{R_2 R_3}{R_4}$ $R_2 = \frac{R_1 R_4}{R_3}$ $R_3 = \frac{R_1 R_4}{R_2}$ $R_4 = \frac{R_2 R_3}{R_1}$ $Z = \frac{(R_1 + R_2)(R_3 + R_4)}{R_1 + R_2 + R_3 + R_4}$	nevyvážený $U_{AB} \neq 0$ $U_{AB} = U \frac{R_1}{R_1 + R_2} - U \frac{R_3}{R_3 + R_4} = \frac{UR_1}{R_{12}} - \frac{UR_3}{R_{34}} = I_{12} R_1 - I_{34} R_3 = U_{12} - U_{34}$ $I_g = \frac{U_{AB}}{R_g}$ U_{AB} — napětí na galvanoměru I_g — proud galvanoměru R_g — vnitřní odpor galvanoměru	

Na obálkách letošní ročenky „Mikroelektronika“ vás chceme seznámit s tím, jak to vypadalo na výstavě ERA '87 ve Žďáru n. S. Na II. a III. straně obálky vidíte: ① záběr do expozice Severomoravského kraje ② napájecí zdroj pro anténní zesilovač R. Smejkal z Jihlavy ③ třípásmovou reprosoustavu ing. Z. Nováka ze Žďáru ④ expozici Jihomoravského kraje ⑤ přijímač pro amatérská pásma V. Zahradila z Ústí n. L. ⑥ stereověž ing. P. Rušáka z Brezna ⑦ univerzální čítač A. Prokeše z Brna ⑧ alfanumerickou klávesnici M. Šípa z Písné ⑨ modemy ing. V. Havla a kol. z Č. Budějovic ⑩ osobní profesionální počítač ISKRA ze SSSR ⑪ soubor učebních pomůcek pro elektrotechniku kolektivu SOU MH z Kutné Hory ⑫ soubor měřicích přístrojů čtrnáctiletého A. Ročka z Č. Lipy (uprostřed) ⑬ ještě jeden záběr do expozice Severomoravského kraje ⑭ simulační panel pro SAPI-1 — střední úprava vody — kolektivu SPŠ Písek.

PŘIPOJENÍ

TISKÁREN C2111 a C2112

K OSOBNÍMU POČÍTAČI ZX SPECTRUM

Ing. Aleš Kastner, CSc., ing. Pavel Vébr

Sériové tiskací mechanismy Consul typu 2111, 2112 jsou snad nejrozšířenějšími mozaikovými tiskárnami ve výpočetních střediscích v ČSSR. Při správné údržbě mají ostrý, dobře čitelný otisk a s formulářovým zařízením DARO mohou tisknout na skládaný tabulační papír bez kopie i s kopií.

Osobní počítač Sinclair ZX Spectrum je u nás rovněž velmi hojně používán. Existují pracoviště, kde se stal pracovním nástrojem konstruktérů, návrhářů, architektů a jiných pracovníků vývoje. Právě jim je určen tento článek, který popisuje spojení uvedených prostředků výpočetní techniky a některé aplikace i s úpravami dostupného programového vybavení.

Popisovaný adaptér a základní tiskový program vycházejí z řešení uvedeného v [1]. Jde o tiskový program nahrazující původní tiskovou rutinu uloženou v pevné paměti (ROM) počítače. Dále je v [1] popsán adaptér přenášející 7 bitů každého znaku do sériového tiskacího mechanismu Consul 2111-03. Adaptér ani program však nebyl vhodný vzhledem k zamýšleným způsobům jeho použití:

- tisku znaků s diakritickými znaménky na upravené tiskárně Consul 2111-03 (nutný přenos všech 8 bitů znaku),
- plynulému tisku na tiskárně Consul 2112, která nemá start/stop režim,
- automatickému rozpoznání typu tiskárny.

Z tohoto důvodu byl adaptér vyřešen zcela nově, takže obsluhuje tiskárny Consul 2111-03 a Consul 2112 bez úprav i s úpravami. Také tiskový program byl přepracován. Jeho varianta, vhodná pro tisk písmen s diakritickými znaménky byla začleněna do textového editoru TASWORD, který byl rovněž upraven tak, aby dovořil pořizovat a tisknout texty obsahující všechna písmena malé i velké české abecedy včetně některých grafických znaků pro tvorbu tabulek a diagramů. Základní varianta je určena pro použití v jazyku BASIC a byla zařazena také do „data-bázového“ programu VU-file.

Popis adaptéru pro spojení ZX-Spectrum s C2111, C2112

Základní částí adaptéru je, obdobně jako v [1], střadač 8 bitů typu MH 3212. Slouží pro uchování a vysílání dat pro sériový tiskací mechanismus. Vzhledem k tomu, že k přenosu dat je využito všech 8 bitů střadače, není možno pro vytvoření signálu SC (řízení přenosu dat z adaptéru — viz tabulka 1) použít stejný způsob, jako v [1]. Pro tento účel je užit pomocný klopný obvod, který je součástí střadače MH 3212. Vhodným zapojením bylo dosaženo, že bezprostředně po každém zápisu nových dat do střadače je nastaven pomocný klopný obvod, jehož stav se po ukonče-

ní zápisu objeví na výstupu IT střadače. Napětí z tohoto výstupu se přímo využívá jako signál SC. Pomocný klopný obvod je nulován tylem signálu AC — viz tabulka 1.

Zbývající část adaptéru je navržena zcela nově a je tvořena pamětí typu MH 74S287 nebo ekvivalentem, např. sovětským obvodem KR 556 RT4. Obsah této paměti musí vyhovovat těmto logickým rovnicím:

$$\begin{aligned} Y1 &= \bar{A} + \bar{B} + \bar{C} + \bar{D} + E + F \\ Y2 &= \bar{A} + \bar{B} + \bar{C} + \bar{D} + E + G \\ Y3 &= \bar{A} + \bar{B} + \bar{C} + \bar{D} + E + H \\ Y4 &= \bar{A} + \bar{B} + \bar{C} + \bar{D} + E \end{aligned}$$

kde: A, B, C, D, E, F, G, H jsou adresové vstupy paměti, Y1, Y2, Y3, Y4 jsou informační výstupy paměti (musí být s otevřeným kolektorem).

Obsah paměti zajišťuje, že adaptér pracuje jako vstup-výstupní brána, v jejíž adrese musí být A0 = H, A1 = L a A3 = H. Tomuto požadavku vyhovuje adresa 9, anebo lépe adresa 253, která je v systému ZX-Spectrum vyhrazena pro tiskárnu. Pro adresování adaptéru lze využít i adresního bitu C paměti PROM, na který je v tomto provedení připojen signál M1 řídící sběrnice. Tento signál měl původně sloužit k rozlišení operace vstup-výstup od cyklu přerušení, při kterém signál IORQ, stejně jako při vstup/výstup operacích, nabývá úrovně L. Podle stavu adres sběrnice v okamžiku přerušení potom hrozí nebezpečí, že vstupně výstupní kanál reaguje na přerušování jako na vstupně výstupní operaci. Vzhledem k důslednému použití obou signálů RD i WR není však využití signálu M1 nutné a lze tedy příslušný adresový vstup paměti PROM použít pro adresování adaptéru (nejlépe vyjmutím kontaktního pera sběrniceového konektoru adaptéru a drátovou propojku na plošném spoji adaptéru).

Vstupní brána, která vznikla naprogramováním paměti PROM, umožňuje přenos tří bitů informace ze sériového tiskacího mechanismu. Z toho je jeden bit použit k přenosu signálu AC do počítače. Po provedení instrukce IN se tato informace objeví v bitu D7. Další bit

Tabulka 1: Zapojení a signály konektoru Consul 2111, 2112

Špička konektoru	Signál STM	Typ STM	Aktivní úroveň	Název signálu
O	AO		L	Připravenost STM přijímat data
J	SO		L	Připravenost RJ přijímat data
M	AC		H	Řízení přenosu dat ze STM
K	SC		H	Řízení přenosu dat z RJ
A až H	SI-1 až SI-8		H	Kódová kombinace (data z RJ)
S	SP		H	Paritní bit z RJ
T	SI-9		H	Platnost paritního bitu z RJ
Q	SI-10	*	H	Konec bloku informací z RJ
R	SI-11		L	Signál nulování RJ
W	SI-12	+	H	Režim práce z RJ
U	AI-1		H	Chyba parity ze STM
Z	AI-2		H	Provádění řádku ze STM
Y	AI-3		H	Chyba mechaniky ze STM
X	AI-4	*	L	Autonomní režim ze STM
I, L, N	OV			Vztažný potenciál 0V
P, V	OV			Vztažný potenciál 0V

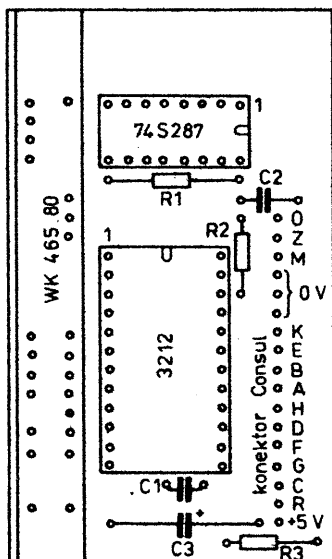
Typ STM: + — pouze Consul 2111
* — pouze Consul 2112

(D1) je použit k přenosu signálu AO (připravenost tiskárny k provozu). Zbývající bit (D4) nemá pevné určení a lze jej použít libovolně např. k přenosu signálu AI-2 (provádění řádku — viz tabulka 1), což umožňuje počítat skutečný počet řádků, ale není vyloučen ani přenos jiného signálu (přepojením na konektor k tiskárně). Schéma adaptéru je na obr. 1.

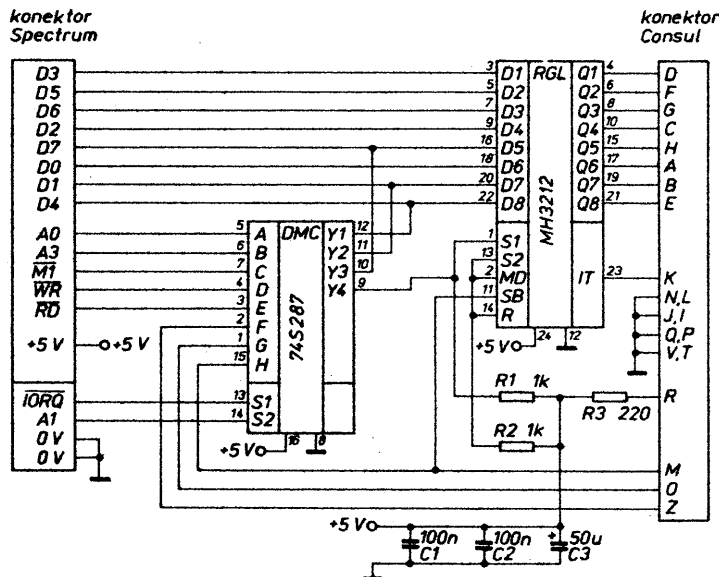
Adaptér je umístěn na destičce s oboustrannými plošnými spoji o rozměrech 43x73 mm. Do otvoru v desce je ze strany spojů zasunut a zapájen konektor typu WK 465 80 zkrácený na 28 párů kontaktů. Typ konektoru je nutno dodržet, neboť velmi podobný konektor WK 180 18 má poněkud odlišnou rozteč kontaktů. Nepoužitá kontaktní pera jsou vyjmuta, takže kontaktní pera zůstávají na těchto pozicích:

A3, A6, A7, A9, A10, A11, A12, A21, A22, A23, A24, B3, B6, B7, B8, B9, B10, B11, B12, B17, B18, B19, B24.

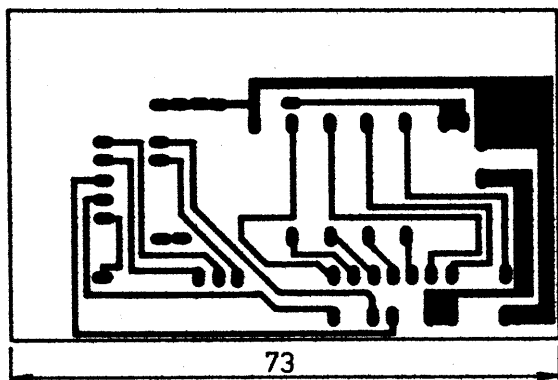
Konektor je podložen vhodnou di-
stanční podložkou, např. destičkou



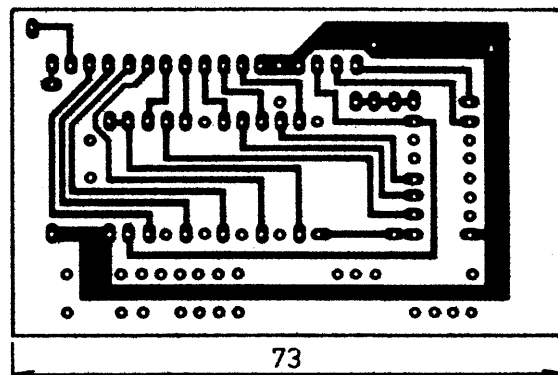
Obr. 2b. Rozložení součástek na desce s plošnými spoji W508 interfejsu ZX Spectrum — Consul



Obr. 1. Interfejs ZX-Spectrum — Consul 2111-03/2112



Obr. 2a. Obrazec plošných spojů interfejsu ZX Spectrum — Consul na desce W508



tloušťky zhruba 3mm s otvory pro vývody kontaktních per, a zasunut na doraz. Před zapájením je vhodné kontaktní pera ohnout k destičce pro mechanické zpevnění. Pro spojení s tiskárnou je použit plochý mnohožilový vodič PNLV 20x0,15, z něhož je použito 17 vodičů. Délka spojovacího vodiče nesmí přesáhnout 3 m. Vodič je upevněn dvojím protažením podélným otvorem v horní části destičky a je zakončen konektorem Consul.

Základní varianta tiskového programu

Základní tiskový program je určen pro tisk znaků kódu ASCII modifikace Spectrum včetně tisku klíčových slov jazyka BASIC. Program lze využít k tisku příkazy LPRINT (PRINT #3) a LLIST. Nelze tisknout obsah obrazovky (příkaz COPY). Znakový repertoár tiskáren Consul se liší od kódu ASCII-Spectrum v těchto znacích:

— „dolar“ (24h) je nahrazen měnovým znakem „\$“,
— „libra“ (60h) znakem „£“.

— „šipka vzhůru“ (5Eh) znakem „↑“,
— „vinovka“ (7Eh) znakem nadtržení „~“,
— „copyright“ (7Fh) je nahrazen mezerou.

V kódu ASCII-Spectrum je několik služebních znaků, které mají při tisku tento význam:

— 06h (čárka v PRINT) — interpretuje se jako tabulace na nejbližší zarážku — sloupce 1, 17, 33, 49, ...
— 09h (BACKSPACE) — zpět o znak (jen C2111),
— 0Ah (CHR\$ 10) — řádkování (jen C2111),
— 0Bh (CHR\$ 11) — vertikální tabulace (jen s formulářovým zařízením DARO 1160),
— 0Ch (CHR\$ 12) — posun na začátek nové stránky,
— 0Dh (ENTER nebo apostrof v PRINT) — nový řádek,
— 12h, 13h, 14h (FLASH 1/0, BRIGHT 1/0, INVERSE 1/0) — změna barvy (jen C2111) 1-červená, 0-černá,
— 17h (TAB n) — posun na sloupec určeného čísla,
— 1Bh (CHR\$ 27) — zpětný řádek (jen C2111 bez DARO 1160).

Není vhodné tisknout texty s ukazateli AT na tiskárně. Tento atribut tiskový program zcela ignoruje, právě tak jako atributy barev. Kódy od 00h do 1Fh

neuvedené v tomto výčtu jsou nahrazeny znakem „?“ Kódy od 80h do A4h jsou nahrazeny znakem „.“.

Tiskový program obsahuje část inicializační a část výkonnou. Inicializační část nastaví způsob stránkování, počty řádků na stránce a znaků na řádku, nuluje počítadla a při prvním spuštění určí typ tiskárny podle délky odezvy na znak „návrat vozu“. Způsob stránkování si určí uživatel podle toho, zda hodlá tisknout na jednotlivé listy nebo rolovaný či skládaný tabulační papír a má-li tiskárnu opatřenou formulářovým zařízením DARO.

Inicializace se spustí příkazem jazyka BASIC

RANDOMIZE USR a+x,

kde: a je počáteční adresa programu (v základní verzi 64200),

x je 0, má-li program počítat řádky a stránkovat, 50, stránkuje-li pilotní páska zařízení DARO 1160, 100, bude-li se tisknout na jednotlivé listy.

Počet znaků na řádku lze zadat příkazem

POKE a+10,n (standardně n=128).

Pokud má program počítat řádky (x je 0 nebo 100), je počet řádků na stránce možno změnit příkazem

POKE a+11,n (standardně n=68).

Počet řádků vynechávaných na konci stránky (přeskočení přehybu) se dá změnit příkazem

POKE a+12,n (standardně n=4).

Poslední dvě hodnoty dávají v součtu celkový počet řádků na stránce (nejčastěji 72), který musí být při změně dodržen. Po změně kterékoliv z těchto hodnot je nutná nová inicializace. Není-li tiskárna při inicializaci zapnuta, ozve se akustický signál, který trvá tak dlouho, dokud uživatel tiskárnu nezapne nebo dokud nestiskne BREAK.

Výkonná část programu interpretuje každý znak a tisknutelné znaky ukládá do vyrovnávací paměti. Po příchodu znaku s kódovou kombinací 0Dh se obsah této paměti vytiskne na jeden řádek. Při tisku se vynechávají pravostranné mezery. Uvedený způsob tisku dovoluje požit Consul 2112 a navíc zkracuje dobu tisku.

Znak 0Ch (nová stránka) se interpretuje okamžitě. Je-li nastaven režim tisku na jednotlivé listy, nebo přijde-li tento znak, je po odpočítání stanoveného počtu řádků na stránce vydán akustický signál, tisk se zastaví a operátor může vyměnit list papíru v tiskárně. Podržením BREAK lze navíc program zastavit na konci každého řádku. V obou případech se ozve zvukový signál, po němž má uživatel tyto možnosti pokračování:

- klávesou „Z“ zastaví tisk s návratem do systému BASIC a s hlášením „D BREAK — CONT repeats“;
- klávesou „C“ pokračuje v tisku dalším řádkem;
- klávesou „I“ provede inicializaci počítadla, program předpokládá tisk od začátku nové stránky, nutno ještě stisknout klávesu „C“.

Tiskový program ve své základní variantě nepoužívá „PRINT buffer“ na adrese 5B00h. Vyrovnávací paměť je umístěna těsně za všemi instrukcemi programu a s její délkou je třeba počítat. Aby byla oblast „PRINT buffer“ využita, lze změnit poslední řádek programu na

Buf EQU # 5B00

Zdrojový program je určen pro přeladač GENS3 (Hisoft).

Použití tiskového programu v programu VU-file

Program VU-FILE pracuje se soubory záznamů kartotéčního charakteru. Jednotlivé záznamy lze pořizovat, přidávat, měnit a rušit. Formáty zobrazení a tisku lze podle potřeby definovat. Do programu VU-FILE byl přidán tiskový program, který se inicializuje po zavedení komplexu programů do operační paměti. Ke své činnosti využívá tato varianta programu systémovou vyrovnávací paměť tisku na adrese 5B00h.

Úpravy textového editoru Tasword

V repertoáru znaků upravené tiskárny Consul 2111-03 jsou malá i velká písmena české abecedy (s diakritickými znaménky), znaky pro tvorbu tabulek a rámečků a speciální znaky. Česká varianta editoru textu — (TASWORD/Cs) u nás dostupná však obsa-

huje pouze 14 písmen malé české abecedy s diakritickými znaménky a znak pro háček za písmenem. Cílem úprav programů TASWORD bylo:

- umožnit tisk celé české a slovenské abecedy i přehlásek,
- umožnit tisk rámečků a tabulek,
- vkládat uvedené znaky z klávesnice bez použití grafiky,
- doplnit generátor úzkých i širokých znaků pro zobrazení.

Nejprve byla vytvořena speciální varianta tiskového programu s určitými zvláštnostmi. Hlavní změna spočívá v tom, že do tiskárny jsou posílány všechny kódové kombinace 20h až 7Fh a A0h až FFh (a některé další). Nedochozí ke konverzi kódů větších než A4h na klíčová slova jazyka BASIC. Program nepočítá s příchodem tiskových atributů jazyka BASIC s výjimkou tabulátoru a změn barvy, akceptuje však řídicí kódy pro posun formuláře. Při tisku na C2112 převádí písmena s diakritickými znaménky na odpovídající písmena bez nich.

Ve strojové části programu TASWORD byla nejprve změněna rutina pro zobrazení úzkých písmen, což dovolilo zmenšit rozsah generátoru znaků na polovinu a doplnit obrazy potřebných písmen české abecedy a rámečkových znaků v generátoru úzkých znaků a písmen české abecedy v generátoru širokých znaků. Dále byly obnoveny obrazy původních grafických znaků.

Vkládání doplněných znaků z klávesnice je řešeno dvojím způsobem. Písmena s diakritickými znaménky vyžadují obdobný postup jako na psacím stroji, tedy nejprve zvolit znaménko, pak dopsat písmeno bez znaménka. Znaménko se volí jednou ze čtyř „nepsmyšlných“ kombinací kláves ENTER a SPACE a oběma klávesami SHIFT. Navolené znaménko se zobrazí na pozici kurzoru, který se neposunuje. Nyní je další vložený znak testován tabulkou písmen přípustných s tímto znaménkem. Je-li písmeno v tabulce obsaženo, generuje se kód písmene s diakritickým znaménkem, jinak se volba znaménka ruší a písmeno se do textu zaznamená beze změny. Kroužek na „ü“ se volí jako háček.

Grafické znaky pro tabulky a rámečky s ostrými nebo kulatými rohy se generují v rozšířeném režimu TASWORDu po stisku kláves 1 až 8 samotných, nebo s klávesou SYMBOL SHIFT, celkem tedy 16 znaků.

Texty pořizované původní českou verzí programu TASWORD nejsou po zavedení novou verzí čitelné. Proto byla vytvořena konverzní rutina, kterou lze vyvolat po funkcích LOAD/MERGE. Znaky odpovídající grafickým symbolům jsou změněny na příslušná písmena české abecedy v kódu ASCII/Cs.

V části programu TASWORD, psané v jazyku BASIC, byly provedeny rovněž některé úpravy. Po stisknutí „STOP“ vystupuje strojový program do části BASIC a zobrazí se přehled možností. Jestliže si uživatel zvolí tisk textu, dotáže se program na rozteč řádků, číslo prvního a posledního tištěného řádku a způsob stránkování. Prázdná odpověď na poslední dotaz se považuje

je za ANO, tedy stránkování pilotní páskou formulářového zařízení DARO. Je-li zadáno N (NE), pak se nastaví inicializace počítáním řádků. Následuje poslední upozornění před tiskem, že má být nastaven začátek stránky. Pak je teprve provedena inicializace tiskového programu. Není-li tiskárna připravena k tisku, je na obrazovku vypsána zpráva a program čeká na ENTER. Pak se inicializace opakuje.

Během tisku lze program přerušit dvojím způsobem. BREAK zastaví vlastní tiskový program; lze nulovat počítadla (klávesa I) a pokračovat (klávesa C), nebo jen pokračovat. Zastavit tisk tímto způsobem nelze. K tomu je nutno podržet klávesu Q. Oba druhy přerušování účinkují až po dotisknutí řádku; do té doby je nutno zvolenou klávesu (BREAK nebo Q) držet.

Jednou z možností programu TASWORD je definice grafických znaků a kódů řídicích funkcí tiskárny. Po zvolení této funkce se zobrazí současná interpretace grafických znaků s kódy 128 až 143. Každý z těchto znaků může být vyslán do tiskárny jako posloupnost až čtyř kódů pro vykonání některé řídicí funkce. Například změna barvy pro tiskový program se kóduje jako 18,1 — červená, 18,0 — černá, nová stránka jako 12 (čísla jsou dekadická).

Po ukončení těchto definic lze ještě zadat všeobecné řídicí kódy pro tisk. Nulová hodnota (kromě kódu 4) znamená, že kód se nevysílá.

Kód 1 — použil by se u tiskáren, kde kód řídicí funkce musí předcházet vždy stejný znak — ESCAPE. Tiskárny C2111, C2112 jej nepoužívají.
Kód 2 — kód zahájení tisku; pro tiskárnu Consul nemá význam,
Kód 3 — kód ukončení tisku; pro tiskárnu Consul nemá význam,
Kód 4 — je to adresa inicializační rutiny tisku (dekadicky) — nesmí se změnit!

Následující kódy se vysílají tiskárně na konci každého řádku:

návrat vozu (CR) — pro obě tiskárny Consul a tiskový program zastupuje znak „nový řádek“;
nový řádek (LF) — ponechat nulu (měl by význam pro dálhopis).

Poslední údaj se týká odsazení tištěného textu doprava. Protože vyrovnávací paměť řádku v tiskovém programu má kapacitu jen 80 znaků, řádek textu má 64 znaků, je možno uvést nejvýše 16.

Úpravy mozaikových tiskáren C2111-03, C2112

Tiskárnu Consul 2111-03 lze upravit pro tisk písmen české abecedy podle ZN 21/84b, jehož správcem je Dopravní podnik města Brna. Úprava je poměrně složitá a předpokládá dodávku tiskací hlavy obsahující 11 magnetů a jehel od výrobce Zbrojovky Brno, k. p. Úprava představuje souhrn těchto dílčích operací:

1. Výměna a nastavení nové tiskací hlavy.

2. Doosazení výkonových stupňů jehel 10 a 11.
3. Doplnění desky ochran obvody pro jehly 10 a 11.
4. Doplnění nezbytných ověřených spojů v roštu.
5. Výměna paměti PROM v generátoru znaků.

Takto upravenou tiskárnu C2111-03 lze využívat jen programem TASWORD. Pro základní verzi tiskového programu a pro použití tisku programem VU-FILE není úprava potřebná.

Tiskárnu Consul 2112 lze upravit poměrně snadno. Pro tisk malých pís-

men latinské abecedy stačí doplnit tři paměti PROM do desky generátoru znaků (deska 6-140.316, pozice 32 dle [2] — obvod 140.124, pozice 33 — 140.125 a pozice 35 — 140.127). Tyto paměti jsou standardně osazovány do generátoru C2111. Dále je nutno změnit význam signálu SI-9, aby tiskárna nekontrolovala správnost přijatých znaků paritou. Proto je nutno přepojit na desce 1 (vstupní obvody, viz [2]) propojku PR2 z polohy C do polohy D (výkres desky 140.201 popř. 140.301). Po těchto úpravách není tisk ještě bezvadný, protože C2112 má jen 8-jehlovou tiskací hlavu, takže spodní body písmen j, y, g, p a q se netisknou.

Pro dokonalejší tisk by bylo nutno osadit 9-jehlovou tiskací hlavu (běžný náhradní díl) a postupovat obdobně jako při úpravě C2111-03.

Literatura

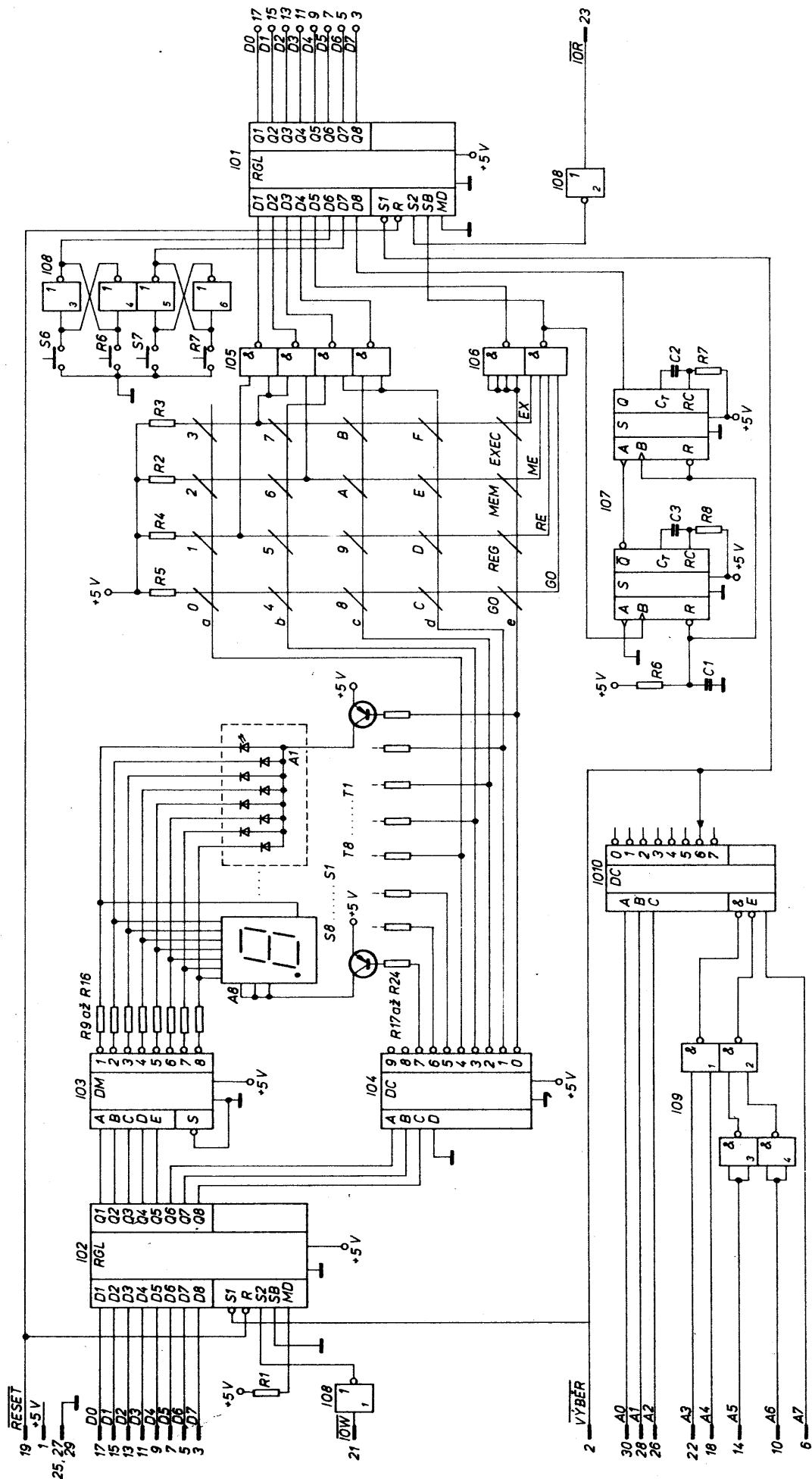
- [1] Pobříslo, J., Pobříslo J.: Popis programu k připojení tiskárny Consul 2111-03 ke SPECTRU. Jablonec n. Nisou, 1985.
- [2] Consul 2111, 2112 — technický popis, IV. vydání, Zbrojovka Brno k. p. 1982

Výpis obslužného programu

```

consul 211.1 - tisk
20 #H consul 211.1 - tisk
30 ;Ekvivalence
40 ;
FAC8 50 Zac EQU 64200
50C5 60 PR_ruta EQU #50C5 ;adr.tisk.rutiny kanalu "p"
09F4 70 PR_OUT EQU #09F4 ;rut.pro klic.slova
03B5 80 BEEPER EQU #03B5 ;piskaci rutina
1F54 90 BR_KEY EQU #1F54 ;test na BREAK
100 #E
110 ;Inicilizace programu
120 ;
FAC8 130 ORG Zac
FAC8 140 Ias LD HL,indas ;inic
FACB 150 LD (HL),#01 ;aut.str.
FACD 160 JR Init ;preskok
FAD2 170 ORG Ias+10
FAD2 180 znr DEFB 128 ;pocet zn./rad.
FAD3 190 rds DEFB 68 ;pocet radku/str.
FAD4 200 rdp DEFB 4 ;pocet radku pres prehyb
FAFA 210 ORG Ias+50
220 #E
FAFA 230 Ips LD HL,indas ;inic.strankovani
FAFD 240 LD (HL),#02 ;pilotni paskou
FAFF 250 JR Init ;---
FB2C 260 ORG Ias+100
270 #E
FB2C 280 Irs LD HL,indas ;inic.rucniho
FB2F 290 LD (HL),#00 ;strankovani
300 #E
FB31 310 Init LD HL,Tisk ;zamena adr
FB34 320 LD (PR_ruta),HL ;PRINT-rutiny kanalu "p"
FB37 330 LD HL,znpoc ;nastav.
FB3A 340 LD A,#01 ;1.znak na radku
FB3C 350 LD (HL),A
FB3D 360 INC HL
FB3E 370 INC HL
FB3F 380 LD (HL),A ;1.radek na strance
FB40 390 DEC HL
FB41 400 LD A,(znr) ;pocet zn/rad
FB44 410 INC A ;+1
FB45 420 LD (HL),A
FB46 430 INC HL
FB47 440 INC HL
FB48 450 LD A,(rds) ;pocet rad/str
FB4B 460 INC A ;+1
FB4C 470 LD (HL),A
FB4D 480 LD HL,Buf ;init pozici
FB50 490 LD (pozbuf),HL ;v bufferu
FB53 500 LD (poz),HL ;pozice nemezery + 1
FB56 510 LD HL,typCon ;A(1typ C211x)
FB59 520 SUB A
FB5A 530 CP (HL) ;urcen ?
FB5B 540 RET NZ ;ano...
FB5C 550 tonl IN A,(#09) ;cti
FB5E 560 BIT 6,A ;plati ?
FB60 570 JR Z,tonl ;ne
FB62 580 BIT 1,A ;ne
FB64 590 JR Z,online
FB66 600 PUSH HL
FB67 610 LD HL,#03E8
FB6A 620 LD DE,#0258
FB6D 630 CALL BEEPER ;zvuk
FB70 640 CALL BR_KEY ;test BREAK
FB73 650 POP HL
FB74 660 JR C,tonl ;ne
FB76 670 RET
FB77 680 online DI
FB78 690 LD B,4
FB7A 700 cyka LD A,#20 ;mezera
FB7C 710 CALL Tiskzn
FB7F 720 CALL cyka ;4 x
FB81 730 LD B,2
FB83 740 cyknv LD A,#0D ;navrat voziku
FB85 750 CALL Tiskzn
FB88 760 DJNZ cyknv ;2 x
FB8A 770 LD B,12
FB8C 780 vnc PUSH BC ;12 x 200
FB8D 790 LD B,200
FB8F 800 DJNZ v ;pockej
FB91 810 POP BC
FB92 820 DJNZ vnc
FB94 830 Rtest IN A,(#09)
FB96 840 BIT 6,A ;plati ?
FB98 850 JR Z,Rtest ;ne
FB9A 860 BIT 7,A ;ready ?
FB9C 870 JR Z,jeto11 ;ne
FB9E 880 LD A,2 ;C2112
FBA0 890 JR Cspol
FBA2 900 jeto11 LD A,1 ;C2111
FBA4 910 Cspol LD (HL),A ;uloz
FBA5 920 EI
FBA6 930 RET ;navrat
940 #E
950 ;Promerme
960 ;
FBA7 970 znpoc DEFB #00 ;pocitadlo znaku na radku
FBA8 980 znr DEFB #00 ;pocet znaku na radku + 1
FBA9 990 radpoc DEFB #00 ;pocitadlo radku na str.
FBAA 1000 radstr DEFB #00 ;pocet radku na str.
FBA8 1010 pozTAB DEFB #00 ;nova pozice TAB
FBAC 1020 indas DEFB #00 ;aut.strankovani - 1
FBAD 1030 typCon DEFB #00 ;typ C211x
FBAE 1040 pozbuf DEFB 0 ;pozice v bufferu
FBB0 1050 pozr DEFB 0 ;pozice posli.nemezery+1
1060 #E
1070 ;Hlavni program
1080 ;
FBB2 1090 Tisk LD HL,IGN
FBB5 1100 DEC HL ;ignorovat ?
FBB6 1110 RET NZ ;ano
FBB7 1120 INC HL
FBB8 1130 LD HL,typCon
FBBB 1140 LD C,(HL) ;id.Consulu
FBBE 1150 CP #A5 ;klic.slovo ?
FBBF 1160 JP NC,PR_OUT ;ano...
FBC1 1170 CP #80 ;znaky #80-#A4 ?
FBC3 1180 JR NC,K80_A4 ;ano...
FBC5 1190 CP #20 ;tisknutelne ?
FBC7 1200 JR NC,TiZn ;ano...
FBC9 1210 CP #06 ;carka v PRINT ?
FBCB 1220 JP Z,PrCom ;ano...
FBCE 1230 CP #09 ;posun zpet ?
FBD0 1240 JR Z,Kzpet ;ano
FBD2 1250 BIT 1,C ;C2112 ?
FBD4 1260 JR NZ,c2a ;ano
FBD6 1270 CP #0B ;vert.tabelace ?
FBD8 1280 JR Z,TiZn ;ano
FBD9 1290 c2a CP #0C ;Page?
FBD0 1300 JP Z,Kpage ;ano...
FBD1 1310 CP #0D ;ENTER ?
FBE1 1320 JR Z,Kenter ;ano...
FBE3 1330 CP #10 ;nepouzite kody ?
FBE5 1340 JR C,Knetis ;ano...
FBE7 1350 BIT 1,C ;C2112 ?
FBE9 1360 JR NZ,c2b ;ano
FBEB 1370 CP #1B ;ESC-zpet.radek?
FBED 1380 JR Z,TiZn ;ano
FBEE 1390 CP #12 ;FLASH ?
FBF1 1400 JP C,At1 ;ano...INK a PAPER
FBF4 1410 CP #15 ;OVER ?
FBF6 1420 JP C,At2 ;ano...FLASH,BRIGHT,INVERSE
FBF9 1430 c2b CP #16 ;AT ?
FBFB 1440 JP C,At1 ;ano...OVER
FBFE 1450 JP Z,AT ;16-AL
FC01 1460 CP #18 ;#18 ?
FC03 1470 JP Z,TAB ;ano...TAB
FC06 1480 JR Knetis ;nepouzite..."?
1490 #E
FC08 1500 Kenter LD HL,znpoc ;ne...radku a navrat
FC0B 1510 JR Nrad
FC0D 1520 Kzpet LD A,#08 ;zpet
FC0F 1530 BIT 1,C ;C2112?
FC11 1540 RET NZ ;ano->
FC12 1550 JR TiZn ;->
FC14 1560 Knetis CP #00 ;NULL ?
FC16 1570 RET Z ;ano...
FC17 1580 LD A,#3F ;otaznik
FC19 1590 JR TiZn ;->
FC1B 1600 K80_A4 LD A,#2E ;tecka
FC1D 1610 TiZn CALL Dobufr ;-> tisk
FC20 1620 LD HL,znpoc
FC23 1630 LD A,(HL)
FC24 1640 INC A ;zvys poc.zn/rad
FC25 1650 LD (HL),A
FC26 1660 INC HL
FC27 1670 CP (HL) ;koniec radku ?
FC28 1680 RET NZ ;ne...
FC29 1690 DEC HL

```

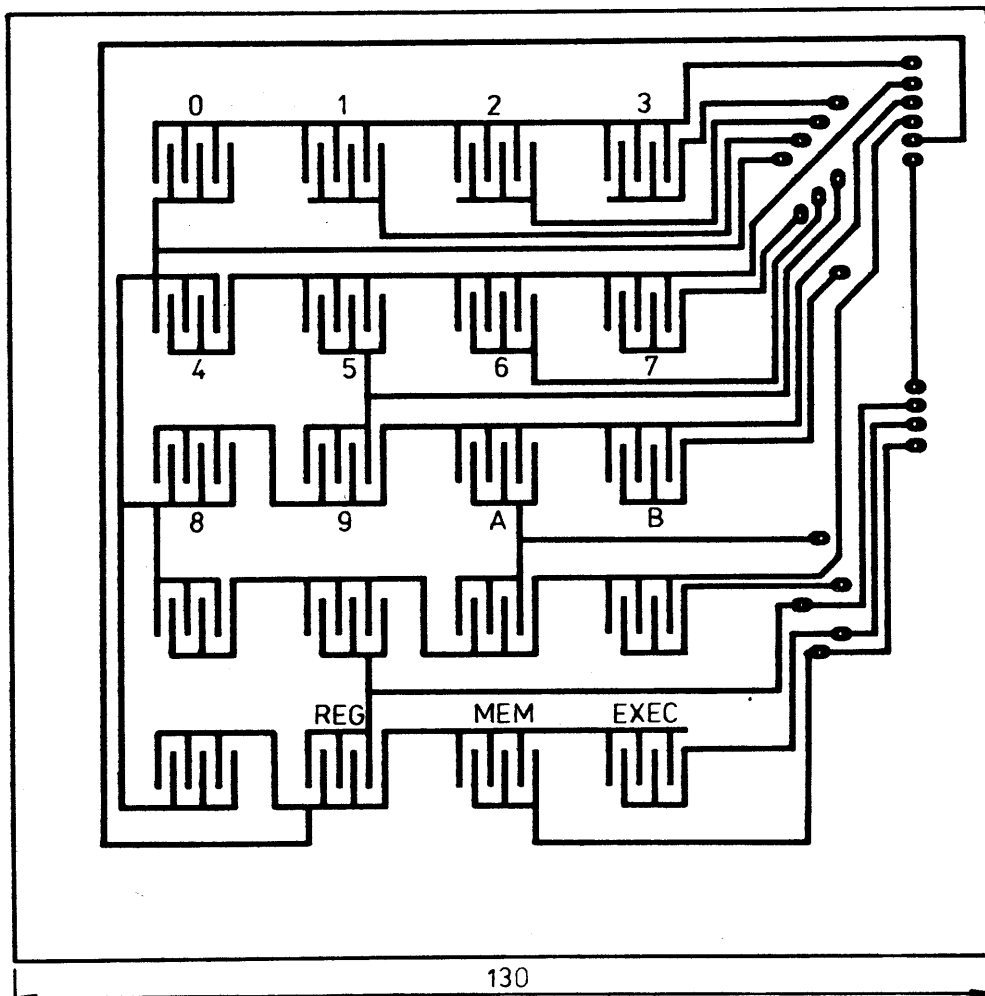
Obr. 1. Schéma mikroterminálu MT-80

Obr. 2.
Obrazec plošných
spojů na desce
membránové klávesnice
W509 mikroterminálu
MT-80

Seznam součástek

IO1, IO2	MH3212
IO3	MH74188
IO4	MH7442
IO5	MH7400
IO6	MH7420
IO7	74123
IO8	MH7404
IO9	MH7400
IO10	MH3205
S1 až S8	LQ400
T1 až T8	KF517
R1	1 kΩ, TR161
R2 až R5	1,5 kΩ TR161
R6	1 kΩ TR212
R7, R8	10 kΩ TR161
R9 až R16	330 Ω TR212
R17 až R24	3,3 kΩ TR212
C1	100 nF
C2	68 nF
C3	150 nF

všechny nejlépe keramické



to, že vyžaduje použití desky plošných spojů vysoké přesnosti nejlépe s provovenými dírami. Intelka je naopak nevýhodná v tom, že pro vytvoření minimálního uspořádání mikropočítače jsou nutné nejméně dvě desky: základní jednotka a paměť. JMP-80 je na desce rozměru 170 x 270 mm zhotovené v běžné třídě přesnosti a na základní desce obsahuje základní jednotku, paměť EPROM 4 kB a RAM 4 kB i obvody vstupu i výstupu. Sběrnice je podobná jako u Intelky, avšak využívá konektory FRB. Myslím, že po obsáhlem článku v AR B1 a B2/83 nemá smysl tento počítač blíže popisovat. Případným zájemcům předlohu pro výrobu plošného spoje zapůjčím.

Nyní zpět k mikroterminálu. Schéma zapojení je na obr. 1. Skládá se z obvodů klávesnice, obvodu displeje a obvodu adresování. Adresu mikroterminálu je možné v určité oblasti volit vhodným zapojením drátových propojek do adresového pole. Takto volíme skupiny po osmi adresách. Výslednou jedinou adresu volíme propojkou na výstupech IO10. Volba adres propojkou je uvedena v tab. 1. Obvody IO9 a IO10 lze vynechat a mikroterminál adresovat přímo ze systému signálem Výběr na špičce konektoru č. 2. K oddělení vstupu a výstupu od datových sběrnic systému a výstupu od datových sběrnic systém slouží IO1 a IO2. Signál z datové sběrnic za IO2 se dekoduje na seg-

Tab. 1.

oktet	A	B	C	D	E	F	G
13	3	1	5	6	2	4	7
14	1	4	5	6	2	3	7
15	3	4	5	6	2	—	7
19	3	4	2	6	7	—	5
21	3	1	5	2	7	4	6
22	1	4	5	2	7	3	6
23	3	4	5	2	7	—	6
25	3	1	2	6	7	5	4
26	1	4	2	6	7	5	3
27	3	4	2	6	7	—	5
28	1	2	5	6	7	4	3
29	3	2	5	6	7	—	4
30	1	4	5	6	7	3	—
31	3	4	5	6	7	—	—

menty a pozice v dekodech IO3 a IO4. IO3 je třeba naprogramovat podle požadavků na tvar zobrazovaných znaků. Je možné zobrazit nejen číslice, ale i některé jednoduché znaky. Pro segmenty, které mají svítit, je třeba naprogramovat do MH74188 hodnotu L. Výstupy IO4 určují pozici segmentového zobrazovače, na které je právě zobrazovaná sestava segmentů. Displej pracuje v multiplexním režimu. Logická nula, která cyklicky spíná jednotlivé pozice displeje, je využita i pro klávesnici. Signály klávesnice jsou kódovány tak, že výstup pro tlačítka 0 až F je přímo v kódu BCD. Bit D5 charakterizuje zbývající čtyři funkční tlačítka. Hodnotu bitů D6 a D7 je možné navolit pomocí obvodů RS z IO8. Logická jednotka na D8 charakterizuje platná

data. Po špatných zkušenostech s klávesnicí PMI-80 byl vliv zákmitů tlačítek odstraněn technickými prostředky přímo v mikroterminálu. Data se do mikropočítače čtou až po ustálení stavu tlačítek. Klávesnice je membránová, podobná jakou má mikropočítač JPR. Postup výroby je popsán v AR B2/83.

Několik poznámek ke konstrukci

Mikroterminál je na desce s plošnými spoji o rozměru 170 x 130 mm. Těsně nad touto deskou je pomocí rozpěrek upevněna sestavená membránová klávesnice, která je propojena se základní deskou drátovými propojkami. K mikropočítači se připojuje mikroterminál přes konektor s propojovacím kabelem. Segmentovky je vhodné umístit do objímky, aby se dostaly do úrovně klávesnice. Rezistory R2 a R5 jsou buď na opačné straně plošného spoje nebo nad IO6. Zapojení samo v sobě neskrývá žádné záludnosti. S vhodným programem pracuje podle požadavků uživatele.

Prerušovací systém a časovač mikropočítačov

ing. Š. Šadlák, ing. P. Chovan

Príspevok je určený pre užívateľov mikropočítačov neobsahujúcich prerušovací systém popr. časovač a hlavne pre záujemcov, ktorí nemajú možnosť zaobstarať si obvody MH3214, MH3212 a 18253, používané pre tento účel v priemyselne vyrábaných systémoch. V predkladanom riešení sú použité dostupné integrované obvody TTL malej integrácie a čítače MH74193 (74192). Riešenie predstavuje rozšírenie možného jednéhoúrovňového prerušovania na viacúrovňový prerušovací systém a ďalej obsahuje jeden 8bitový časovač, pričom výstup časovača zároveň tvorí jednu prerušovaciu úroveň. Zapojenie umožňuje rozšíriť funkciu mikropočítača pre riadenie v reálnom čase a realizovať časovacie funkcie bez použitia programových časovacích slučiek. Aj keď je možné riešenie využiť všeobecnejšie, ovládacie signály a funkcie sú popísané k použitiu so školským mikropočítačom PMI 80.

Popis obvodového riešenia prerušovacieho systému

Návrh obvodového riešenia prerušovacieho systému predpokladá existenciu jednéhoúrovňového prerušovacieho systému v mikropočítači. Hardwarovo je táto možnosť zabezpečená tým, že výstup INTA systémového budiča 8228 je cez rezistor 1 k Ω pripojený na napätie +12 V. Po prijatí požiadavky na prerušenie cez vstup INT mikroprocesora 8080, generuje obvod 8228 kód prerušenia RST 7, následkom čoho mikroprocesor vykoná skok na adresu 38 H s uložením návratovej adresy do zásobníka. Od adresy 38 H je v bunkách pamäti (väčšinou ROM) uložená inštrukcia skoku na bunku v pamäti RAM situovanú v oblasti zásobníka. Na bunky od tejto adresy vkladá užívateľ inštrukciu ďalšieho skoku na adresu, kde začína vlastný obslužný program prerušenia.

Obvodové riešenie prerušovacieho systému predstavuje rozšírenie na štvorúrovňový systém, umožňujúci rozlíšiť a spracovať požiadavky na prerušenie: INTR 1 až INTR 4. Štvorúrovňový systém bol zvolený s ohľadom na predpokladanú praktickú potrebu, princíp umožňuje jednoduchým spôsobom rozšírenie až na osemúrovňový systém, prirodzene s väčším počtom potrebných obvodov. Žiadosti o prerušenie INTR 1 a INTR 2 sa privádzajú z riadeného objektu na hodinové vstupy D klopného obvodu IO1. Klopný obvod slúži ako pamäť žiadostí. Zápis žiadostí sa uskutočňuje pri zmene z L do H úrovne signálu žiadostí. Zápis žiadostí je možné blokovať pomocou D vstupov IO1, vtedy je potrebné, aby blokovacie signály DI1 a DI2 mali úroveň L. Riadenie úrovne týchto vstupov a tým akceptovanie popr. zákaz žiadostí je možné zabezpečiť ich spojením s výstupmi brán obvodu 8255 ovládaných programom. Ďalšia prerušovacia úroveň je odvodená od činnosti časovača. Po ukončení časovacej aktivity je výstupným signálom z čítača preklopený D klopný obvod IO7/B, z ktorého výstupu je odoberaný signál

žiadosti o prerušenie INTR 3. Pomocou vstupu D13 nastaveného na úroveň L je možné blokovať zápis tejto žiadosti. Poslednou prerušovacou úrovňou je žiadosť INTR 4, nevyžadujúca prídavnú pamäť. Generovania žiadostí ako aj jej kvitovanie popr. zákaz je zabezpečené jednou z brán obvodu 8255 pracujúcej v móde 1 popr. 2 (strobovaný vstup-výstup) a pomocou riadiacich signálov brány C tohto obvodu. Signály žiadostí na prerušenie zapísané v pamäti a trvajúce až do ich kvitovania, sa logicky sčítajú pomocou obvodu IO2. Výsledný signál, tj. žiadosť o prerušenie INT s aktívnou úrovňou L, je privedený na systémový vstup INT_{EX} mikropočítača PMI 80 alebo po jeho invertovaní priamo na vstup INT mikroprocesora 8080 v iných systémoch.

Pretože sa jedná o rozšírenie jednéhoúrovňového systému, je potrebné zabezpečiť rozlíšenie jednotlivých prerušovacích úrovní. Rozlíšenie je zabezpečené vykonaním operácie vstupu (inštrukciou IN port) zo zvláštného portu. Tento port generuje kód podľa aktivovaných prerušovacích úrovní a tvorí obvod IO4. Na vstupy IO4 sú privedené zapamätané žiadosti o prerušenie, výstupy IO4 sú spojené s linkami DB0 až DB4 dátovej zbernice „vytiahnutými“ pre účel čítania cez rezistory 22 k Ω na úroveň H. Pri inštrukcii čítania je aktivovaný adresový bit A7. Priradením jednotlivých úrovní k linkám DB0 až DB4 a tvarom krátkeho rozlišovacieho programu je stanovená priorita prerušovacích úrovní. Príklad jednoduchého rozlišovacieho programu je uvedený v ďalšej časti. Podľa tohto zapojenia a programu je priorita prerušovacích úrovní nasledovná: INTR 4 (najvyššia priorita), INTR 3, INTR 2, INTR 1. Prípad, že program neidentifikuje ani jednu úroveň, znamená poruchovú činnosť (popr. stlačenie tlačítka INT v mikropočítači PMI 80, čo možno využiť ako ďalšiu prerušovaciu úroveň). Vzhľadom na obvodovú jednoduchosť „kódera“ priorit a identifikačný program je potrebné otázku priorit prerušenia upresniť nasledovne. Uvádzaná priorita úrovni sa realizuje v okamžiku výskytu zapamätaných žiadostí, obsluhovaná začína byť najvyššia priorita. Ak v obslužnom programe tejto úrovne kvitujeme (zrušíme) túto žiadosť a povolíme prerušenie inštruk-

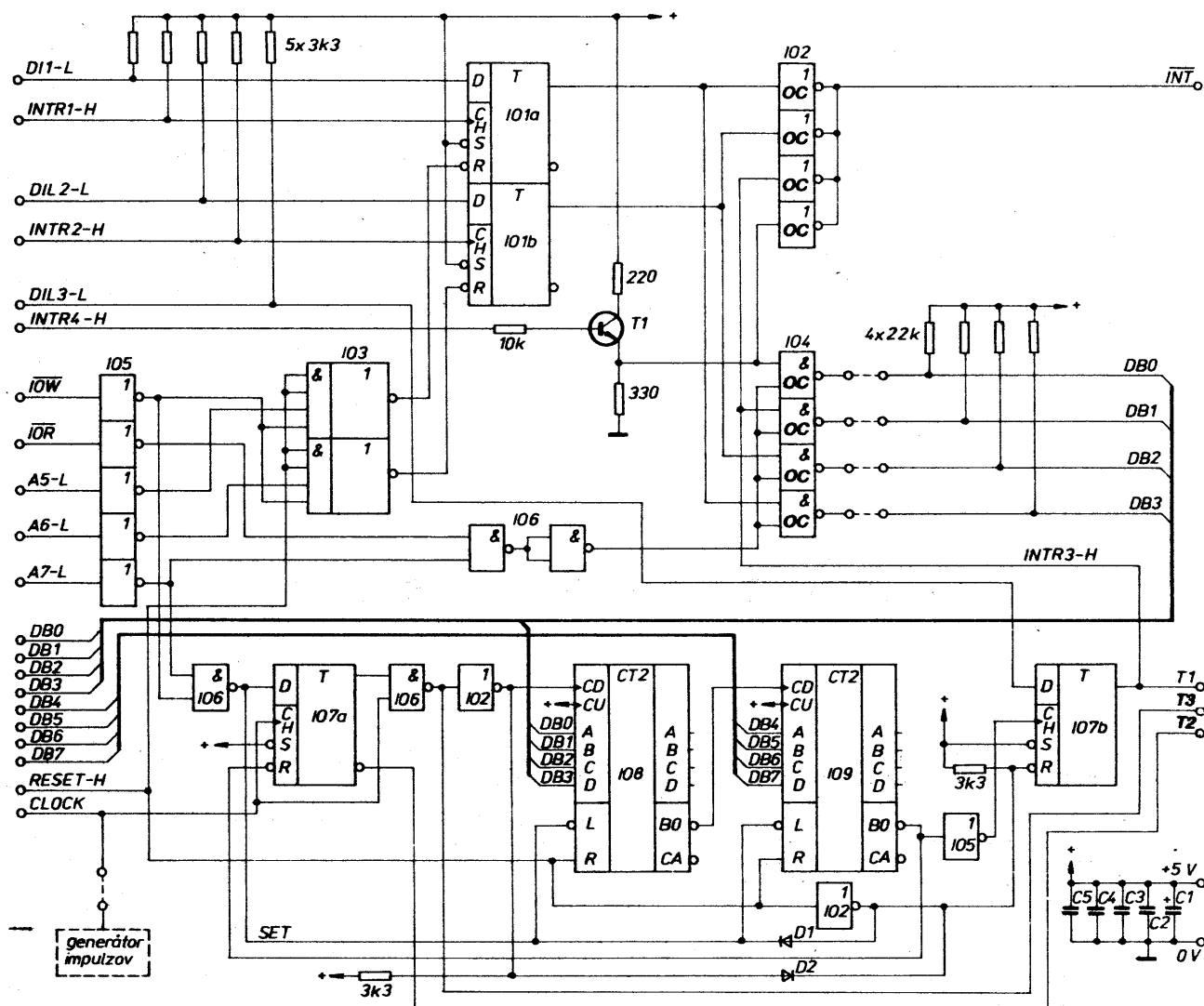
ciou EI, je vyvolané nové prerušenie s obslužným programom najvyššej priority zo zostávajúcich požiadaviek. Takto je možné prerušiť aj vyššiu úroveň nižšou, ak to v obslužných programoch pripustíme. Kvitovanie (zrušenie) žiadosti o prerušenie je generované programovo, konkrétne inštrukciou výstupu (OUT port) do portu pamäti žiadosti. Táto inštrukcia býva v tele obslužného programu tejto žiadosti. Žiadosť INTR 1 je zrušená vykonaním inštrukcie OUT do portu tvoreného IO1/A, pričom je aktívny adresovací bit A5. Obdobne je kvitovaná žiadosť INTR 2, nulovaný je obvod IO1/B a aktivovaný je adresovací bit A6. Na obsahu registra A pri vykonávaní inštrukcie OUT v oboch prípadoch nezáleží. Kvitovanie žiadosti INTR 3 je zabezpečené vykonaním inštrukcie OUT do portu tvoreného IO7/B, aktivovaný je adresovací bit A7, akumulátor musí mať pri vykonávaní inštrukcie nulovú hodnotu. Táto požiadavka vyplýva zo súčasného využívania bitu A7 pre časovač. Kvitovanie žiadosti INTR 4 prebieha podľa pravidiel komunikácie s obvodmi 8255. Použitie adresovacích bitov A5, A6, A7 vychádza z použitia princípu lineárnej selekcie v adresovaní portov a zo skutočnosti, že tieto bity sú v mikropočítači PMI 80 voľné. Po doplnení základnej verzie PMI 80 o užívateľský obvod 8255 a obvodov podľa tohto článku zostáva užívateľovi k dispozícii voľný adresovací bit A4 a adresovacie bity A5, A6 možno použiť pre vstupné porty. Úplný kód adres portov môže byť ovplyvnený pripojením iných portov I/O. Jeho stanovenie ponecháme užívateľovi podobne ako aj pridelenie iných priorit prerušenia podľa aplikácie mikropočítača a povahy riadeného objektu.

Poznámka k nevyužívaniu niektorých úrovní: nevyužívané žiadosti — vstupy INTR 1 INTR 2 a INTR 3 a ich blokovacie vstupy DI1 a DI2 netreba ošetrovať. Ak sa nevyužíva úroveň INTR 3 ale je využívaná časovacia funkcia, je potrebné vstupu D13 priradiť úroveň L.

Rozlišovací program a príklad obslužného programu.

Spôsob vyvolania rozlišovacieho programu ako aj stanovenie priorit úrovní bolo popísané v predchádzajúcej časti. Program je zapísaný v mnemokóde MHB8080 a so symbolickými adresami.

* Uloženie obsahu registra A a príznačného registra do zásobníka z dôvodu využívania týchto registrov v rozlišovacom programe. Inštrukcia spätného vyzdvihnutia ich obsahu POP PSW je obsiahnutá v každom podprograme žiadostí prerušenia.



Obr. 1. Prerušovací systém a časovač mikropočítača

Popis obvodového riešenia časovača

Časovač umožňuje generovať jednu časovú aktivitu nezávisle od činnosti CPJ (po začatí čítania vyvolanom CPJ). Jedná sa v podstate o 8bitový čítač zapojený ako vratný s možnosťou nastavenia začiatkovej hodnoty a postupným odpočítavaním vonkajších hodinových impulzov až do nulovej hodnoty čítača, kedy sa končí časovacia aktivita a môže byť vyvolané prerušenie činnosti CPJ. Časovač pozostáva z dvoch 4bitových čítačov MH74193 (alebo MH74192), vzájomne spojených na 8 bitový vratný čítač a radiačných obvodov. Na nastavovacie vstupy A, B, C, D čítačov sú pripojené linky DB0 až DB7 datovej zbernice. Zápis požadovanej počiatkovej hodnoty času (počtu impulzov) je zabezpečený vykonaním inštrukcie OUT do portu čítačov, pričom je aktivovaný adresovací bit A7. V prípade použitia čítačov 74193 sa jedná o osembitový dvojkový čítač, pri použití čítačov 74192 o dvojitupňový dekadický čítač (do obvodu desiatkového rádu IO9 sa potom ukladá počet desiatok). Rozsah počítania dvojkového čítača činí: 1 až 255 impulzov, desiatkového: 1 až 165 impulzov (je možné nastaviť 15 desiatok).

Pamäť ukončenia časovacej aktivity tvorí D klopny obvod IO7/B, z jeho výstupu sa odoberá signál prerušovacej úrovne INTR 3. D klopny obvod IO7/A a hradlá IO6, IO2 slúžia pre blokovanie počítania a synchronizáciu započítania prvého impulzu. Hradlá IO2 spolu s diodami D1, D2 umožňujú systémové nulovanie (signálom RESET) ako aj funkčné nulovanie sekvenčných obvodov časovača. Z obvodov časovača je možné bezprostredne odoberať časové priebehy T1, T2, T3. Ich priebeh ako aj časová neurčitost vyplýva z obr. 2. Tieto výstupy sú aktivované po každom nastavení čítača, čo v niektorých aplikáciách nie je vhodné (napr. časovanie medzery medzi impulzami). Prípadné potlačenie časovaných výstupov pomocou výstupov obvodu 8255 ponechávame čitateľovi. Pokiaľ sa využíva iba jedna (neopakovaná) časovacia aktivita je pracovná rýchlosť obmedzená iba medznou frekvenciou obvodov časovača. Zložitejšie časové aktivity (opakované s rôznou dobou, popr. dlhé) sa realizujú programovo pomocou výstupných liniek obvodu 8255 a prerušenia vyvolaného čítačom — INTR 3. Oblasť gene-

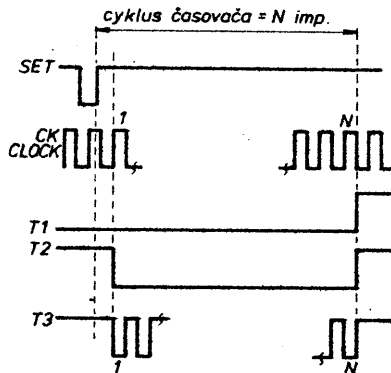
Symbolická adresa	Mnemokód	Vysvetlivka
38 H	JMP NAV 1	Inštrukcia v monitore
NAV 1	JMP NAV INT	Inštrukcia skoku uložená v zásobníkovej časti RAM
INT	PUSH PSW IN port IO4	+ Prečítanie kódu prerušenia z IO4
	RAR JNC NAV INT 4	Skok na obslužný program najvyššej priority
	RAR JNC NAV INT 3	
	RAR JNC NAV INT 2	
	RAR JNC NAV INT 1	
	JMP NAV POR	Skok na program indikácie poruchy
...	...	
INT 1	OUT port IO1/A	Nulovanie KO žiadosti INTR 1
	EI	Povolenie prerušenia
	PUSH	Uloženie obsahu dvojice používaných registrov do zásobníka
...	...	
	POP	Vyzdvihnutie obsahu dvojice používaných registrov
...	POP PSW + RET	
INTR 3	MVI A, D0 OUT port IO7/B	Vynulovanie KO žiadosti INTR3
...	PUSH	
	POP PSW	
	EI	
	RET	

rovaných časov je v tomto prípade obmedzená dobou spracovania obšlužného programu.

Použitie časovača dopĺňujeme v tom zmysle, že neumožňuje generovať súčasne viac časových udalostí, nie je možné prečítať okamžitý stav čítača, počítanie hodinových impulzov možno nahradiť počítaním udalostí z riadeného objektu. Návrh obvodového riešenia generátora hodinových impulzov CLOCK vzhľadom na rôzne potrebné časy ponechávame užívateľovi.

Konštrukčné riešenie

Ku konštrukčnému riešeniu obvodov prerušovacieho systému a časovača iba niekoľko poznámok. Uvedené



Obr. 2. Časové priebehy

obvodové riešenie možno realizovať na doske plošných spojov orientačne s rozmermi 150 x 80 mm. Pre osadenie obvodov podľa obr. 1 spájkovacie pole základnej dosky PMI 80 nepostačuje. Vhodné je vyrobiť zvláštny modul a upevniť ho pomocou upevňovacích skrutiek a distančných stĺpkov nad toto spájkovacie pole. Pre elektrické prepojenie je vhodné vystrojiť modul malými dvoma až troma konektormi FRB. Jeden pre prepojenie so systémovým konektorom PMI 80-K1, druhý pre prepojenie s konektorom PMI 80-K2 a prípadný ďalší pre prepojenie s riadeným okolím. Napájacie napätie (+5 V) možno použiť z PMI 80, kde pri použití napájača SN 080 je dostatočná rezerva v odbere.

Tiskárna k SAPI-1 — snáze

Ing. Miloslav Čapek

Tiskárny jsou jedním z nejpoužívanějších periferních zařízení v mikropočítačových systémech. V mnoha případech je vybavení tiskárnou limitujícím faktorem pro použití v praxi. Výběr těchto zařízení není však na trhu nikterak bohatý. Jednou z možností v této oblasti je nákup relativně dostupné zapisovací jednotky Consul 256.9. Jedná se v podstatě o elektrický psací stroj řízený z externího zařízení pomocí matice ovládacích magnetů. Řídící elektroniku však výrobce nedodává. Tuto funkci zabezpečí popsaný systém.

Jedním z úvodních požadavků bylo vystačit při realizaci s volnými IO vodiči portů desky JPR-1. Proto musely být výkonové spínače doplněny dekodéry. K ovládání tiskárny potom postačí jediný výstupní port (Port 2OUT konektoru X3). Bity 0 až 3 zde tvoří adresu sloupce, v němž leží spínaný magnet, bity 4 až 6 adresu řádku a bit 7 lze ponechat pro ovládání dekodérů adres. Do volných bodů ovládací matice byly pomocí propojek a diod umístěny také ovládací magnety, oddělené na psacím stroji na konektorech K3 a K7. Ze zapisovací jednotky do počítače je snímána také zpětná stavová informace. Není využito všech kontrolních funkcí, a proto postačí jen pět vstupních bitů. Nutným se ukázalo použít vstupní bity z konektoru X2 počítače, asice PORT 0 IN (pro připojenou klávesnici Consul 259.11) nebo PORT 1 IN (pro připojenou klávesnici ANK-1). Obsadíme-li vždy stejné bity, budou se obě řešení lišit pouze adresací vstupního portu (tab. 1). Z uvedeného vyplývá, že koncovky obou kabelů náležejících konektorům X2 a X3 JPR-1 je nutno kontrolními vodiči propojit. To sice není příliš instalačně elegantní, ale je to jednoduché a úsporné. Stykový obvod je na dvou jednoduchých deskách s plošnými spoji (zdroj 5 a 12 V a vlastní obvody), z technologických důvodů jednostranných s množstvím propojek. K desce logiky je přiveden jeden kabel ze SAPI-1 a tři kabely ze zapisovací jednotky.

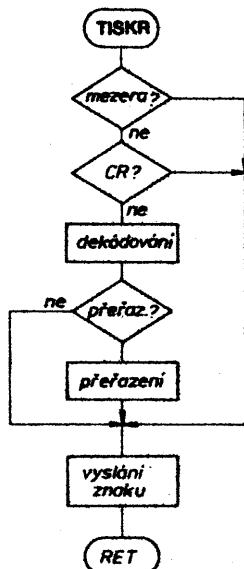
Jednoduché připojení je však třeba doplnit programem. Program

Tab. 1. Adresování podle použité klávesnice

Klávesnice	Obsah na adr. 430E, 436D, 436FH
ANK-1	28H
Consul 259.11	24H

Tab. 2. Doplnkové funkce

Funkce	Realizace
Černá páska	XX=CALL (17248, 48343)
Červená páska	17255, 49183
Tabulátor	17248, 65417
Sazeč	17248, 65409
Vymazávač	17248, 65410
Zpětný krok	17255, 65288



k ovládání tiskárny je napsán ve strojovém kódu a je zabudován do systému ZPS-2 tak, aby bylo možno používat tiskárnu běžnými příkazy BASIC (LLIST, PRINT#L, TAB, SPC). Zajišťuje překódování znaků z kódu ASCII na kód matice magnetů (podprogram CODE), zjistí polohu typového koše u požadovaného znaku a podle potřeby zajistí i přefázení. Zabezpečí také samotné vyslání znaku na PORT 2 s potřebnými časovými parametry (podprogramy VYSL a DELY). Před vysláním znaku kontroluje ukončení všech důležitých funkcí zapisovací jednotky (podprogramem ZNAK). K dekodování používá program tabulku, obsahující kontrolní znaky a velká písmena. V případě, že je třeba i malých písmen, háčeků a čárek, je možno tabulku jednoduše odzadu doplnit. Programem lze ovládat také doplňkové funkce zapisovací jednotky a to volání CALL přes vstupy CTRL 3 a CTRL 4 (tab. 2). Úvodní krátký program INSTL provede instalaci nového ovládacího programu tiskárny do systému.

Program i s tabulkou znaků je kratší než 256 bajtů a vejde se do volné oblasti paměti 4300H až 43FFH. To má velký význam, neboť obslužný program tiskárny bude možno nahraďovat společně s interpretem BASIC. Lze ho ovšem uložit do libovolné oblasti paměti. Je ho třeba zapsat do příslušné oblasti podle výpisu programu buď pomocí příkazu .S monitoru přímo, nebo je možno použít systému TOOL, zapsat zdrojový program přeložit ho a přesunout příkazem M na požadovanou adresu. Podle výpisu obsahu paměti naplníme předepsanou oblast prvky dekodovací tabulky. Návěští T v programu se musí shodovat s adresou prvního z nich. U původního souboru BASIC je třeba dále změnit skok JMP 4407H na JMP 4300H. Skok leží na adrese 4400H. Takto připravený interpreter se nahraje příkazem .KS z oblasti 4300H—5FFFH. Nový BASIC je možno opět startovat příkazem .B. Oproti dřívějšímu přibývá pouze inicializace .A-L=1 před spuštěním.

TESLA Strašnice k. p.

**závod J. Hakena
U nákl. nádraží 6
Praha 3-Žižkov
130 65**

přijme:

**sam. odb. ekonoma-rozboráře
(VŠ nebo ÚSO + praxe)**

**sam. vývoj. pracovníky
(VŠ nebo ÚSO + praxe)**

konstruktéry (ÚSO + praxe)

sam. konstruktéry (VŠ + praxe)

**sam. odb. ekonomy (zásobovače)
(ÚSO + praxe)**

**ved. odb. techn. pracovníka (vedoucí pro-
vozu a údržby) (VŠ + praxe)**



**Zájemci hlase se na osobním oddělení našeho závodu nebo na
tel. 77 63 40.**

**Nábor je povolena na celém území ČSSR s výjimkou vymezeného území.
Ubytování pro svobodné zajistíme v pod. ubytovně. Platové zařazení dle ZEUMS II.**

MIKROELEKTRONIKA

faktor
úspor

a zvyšování účinnosti
lidské práce

TESLA ELTOS oborový podnik zajišťuje technické a obchodní služby v oblasti spotřební a investiční elektroniky všech VHIJ Tesla a také ve vybraných oblastech produkce ostatních odvětví elektrotechnického průmyslu. Plní též úkoly elektronizace národního hospodářství a mezinárodní technicko-obchodní kooperace. Z další rozsáhlé činnosti zajišťuje zejména:

- + Mikroelektronika – vývoj, aplikace, programování, školení a zavádění při elektronizaci národního hospodářství.
- + Dodávky elektronických součástek.
- + Dodávky a servis investičních zařízení, vyšší dodavatelské funkce.
- + Racionalizace a automatizace.
- + Mezinárodní technicko-obchodní kooperace.
- + Průzkumový prodej novinek spotřební elektroniky a elektrotechniky.
- + Prodej a servis spotřební elektroniky s poradenstvím, celostátní zásilková služba.
- + Pomoc radioamatérům a mladým elektronikům, spolupráce se Svazarmem, SSM aj.
- + Multiservis.
- + Průmyslové opravárenství a úpravárenství.
- + Ústřední gesce technického servisu, řízení a kontroly jakosti, zásobování součástkami a náhradními díly.

ZÁVODY S OBLASTNÍ PŮSOBNOSTÍ: v Praze, Ústí nad Labem, Ostravě, Brně, Uherském Brodě, Bratislavě, Banské Bystrici a Košicích.

Účelové závody: Institut mikroelektronických aplikací, Praha (IMA); Dodavatelsko-inženýrský závod, Praha (DIZ); Závod racionalizace a automatizace, Praha (ZAR); Závod průmyslového servisu, regenerace, renovace a kooperace, Týniště nad Orlicí; Závod centrálního zásobování, Uherský Brod.

GENERÁLNÍ ŘEDITELSTVÍ
113 40 Praha 1, Dlouhá 35.

Tel. 231 5396,
dlps 122629

TESLA ELTOS
oborový podnik

ČKD Praha, o. p. závod POLOVODIČE

**na trase metra C
stanice: Mládežnická—Budějovická**

Chcete pracovat v novém, atraktivním prostředí?

Chcete pracovat na nejmodernější výpočetní technice?

Chcete vidět jak se chová vaše technické dílo?

Chcete se podílet na programu automatizace?

Čekáme na vás — informujte se přímo v závodě!

Možnost získání bytu!



PŘIJÍMÁME: programátory, systémové ing., prog.-analytiky, projektanty, teoret. kybernetiky a ing. silnoproudé i slaboproudé elektrotechniky pro vývoj složitých automatických systémů řízení dodávaných do tuzemska i na export.

Přijímáme absolventy všech příbuzných oborů schopné a ochotné se podílet na tomto programu, ať již v oblasti vývoje HW a SW automat. prostředků vyráběných a vyvíjených v ČKD POLOVODIČE, tak v oblasti projektování a návrhů systémů automatizovaného řízení technologických procesů a tech. objektů pro oblast teplých a studených válcoven, hutního a slévárenského průmyslu, cementáren, úpraven rud a dalších.



**Informace: nábor pracovníků —
přímá linka 42 69 65**

ČKD POLOVODIČE, Budějovická 5, Praha 4